BATS Europe
FIX Specification

Version 2.21
27 April 2011

Contents

1 Overview

This document describes BATS Europe's (hereafter, "BATS") interpretation and implementation of the FIX 4.2 specification. BATS uses a subset of the FIX 4.2 protocol for order entry and drop copies. It is assumed that the reader is familiar with the FIX 4.2 protocol as described by the FIX Protocol Organisation.

1.1 Hours of Operation

Refer to the BATS Europe website for hours of operation.

All orders are live upon acceptance by BATS. Orders are rejected if they are received outside the hours BATS is available for trading. BATS does not have an opening or closing auction, however a Market On Close and Trading At Last service is available for liquid stocks. BATS does not support maintaining orders for multiple days (GTC orders). All open orders are canceled on close of the market. Participants will receive an execution report for every open order with *ExecType* (150) = `4` (Canceled).

1.2 Timestamps

All FIX timestamps are GMT as per the FIX standard. Participants are expected to synchronise their clocks with an external time source.

1.3 Symbology

BATS accepts four symbologies: MTF Common Symbology, RIC, ISIN, and SEDOL. Different symbologies may be used on different orders, but it is recommended that Participants use the same symbology for all orders.

If using MTF Common Symbology to identify a stock, the Participant:

- **must** set *Symbol* (55) to the MTF Common Symbology symbol;
- *may optionally* set the *SecurityExchange* (207); and,
- *may optionally* set the *Currency* (15).

If using SEDOL to identify a stock, the Participant:

- **must** set *IDSource* (22) to SEDOL (2);
- **must** set *SecurityID* (48) to the SEDOL;
- *may optionally* set the *SecurityExchange* (207);
- *may optionally* set the *Currency* (15); and,
- *may optionally* set the *Symbol* (55) to the MTF Common Symbology symbol or to the *SecurityID* (48).

If using ISIN to identify a stock, the Participant:

- **must** set *IDSource* (22) to ISIN (4);
- **must** set *SecurityID* (48) to the ISIN;
- **must** set *SecurityExchange* (207) to note the market in which the ISIN trades;
- **must** set the *Currency* (15) field to identify the currency in which the stock is traded; and,
- *may optionally* set the *Symbol* (55) to the MTF Common Symbology symbol or to the *SecurityID* (48).

If using RIC to identify a stock, the Participant:

- **must** set *IDSource* (22) to RIC (5);
- **must** set *SecurityID* (48) to the RIC;
- *may optionally* set the *SecurityExchange* (207);

- *may optionally* set the *Currency* (15) field; and,
- *may optionally* set the *Symbol* (55) to the MTF Common Symbology symbol or to the *SecurityID* (48).

When specifying an optional value as noted above, the value specified must match the value in BATS' symbol database. Otherwise, the order will be rejected.

Execution reports will always respond with the same symbology as was sent in the corresponding New Order Single message.

For additional information about BATS' symbology, see the BATS Europe Market Guide.

1.4 Tick Sizes

The minimum price increment, or tick size, is generally the same as that on the primary market. Tick size is subject to change. Orders entered which violate the tick size will be rejected. Midpoint peg orders are not tick size validated and may execute at a price that is one-half the tick size. Where the mathematical mid price has more than four decimal places then the match mid price will be rounded up to four decimal places.

1.5 Hidden Orders

BATS allows Participants to place hidden orders which are not represented on its market data feed. Hidden orders include pegged orders and orders which have the *DisplayIndicator* (9479) = `I` (Invisible). MiFID regulations require that orders with a notional value less than the Large In Scale (LIS) *must* be displayed unless routed to the BATS Dark Pool[1] or BATS Market On Close[2] or BATS Trading At Last.

BATS will reject any order submitted which violates this regulation. Where no LIS value is specified in the MiFID regulations, Participants may only submit hidden orders with a notional value greater than the value specified by BATS in its reference data unless routed to the BATS Dark Pool or BATS Market On Close or BATS Trading At Last. BATS will reject any order submitted which violates this.

Notional value is calculated differently depending upon order type:

Limit:	*Price* (44) × *OrderQty* (38)
Buy primary peg, sell market peg:	(PBBO bid + *PegDifference* (211)) × *OrderQty* (38)
Sell primary peg, buy market peg:	(PBBO ask + *PegDifference* (211)) × *OrderQty* (38)
Midpoint peg:	(PBBO midpoint + *PegDifference* (211)) × *OrderQty* (38)

More information on the MiFID CESR database can be found at the CESR MiFID website.

1.6 BATS Dark Pool

The BATS Dark Pool is a separate book which allows matching of dark liquidity based at a midpoint reference price. Orders placed into the BATS Dark Pool do not need to be Large In Scale (LIS). BATS Dark Pool orders only interact with other BATS Dark Pool orders. Quotes for BATS Dark Pool orders are not represented on any market data feed. Matches in the BATS Dark Pool are represented as trades on BATS' market data feeds and may be differentiated.

Orders destined for the BATS Dark Pool must be midpoint peg orders[3]. The *RoutingInst* (9303) must be set to one of the following:

`BD`: Routes only to the BATS Dark Pool.

`BA`: Routes to the BATS Dark Pool if midpoint and not LIS, or to the integrated book otherwise. Non-midpoint peg orders may also be sent with BA and will always route to the integrated book.

[1] Field *RoutingInst* (9303) = BD or BA.
[2] Field *ExecInst* (18) = `c` or *OrdType* (40) = `5` or `B`.
[3] *ExecInst* (18) = M.

A limit price may be specified on a BATS Dark Pool order using the *Price* (44) field. If set, execution still only occurs at the midpoint. When the midpoint is a more aggressive price than the limit price, the order will not be available for execution.

1.7 BATS Dark Self Cross

The BATS Dark Self Cross is a separate book which allows Participants to cross only against their own orders at a midpoint reference price. Orders entered for BATS Dark Self Cross do not need to be Large In Scale (LIS). BATS Dark Self Cross orders only interact with other BATS Dark Self Cross orders which have been tagged in the same manner using *CrossFlag* (7740). BATS Dark Self Cross orders are not represented on any market data feed. Matches in the BATS Dark Self Cross are represented as trades on BATS' market data feeds in the same manner as BATS Dark Pool trades.

Orders destined for the BATS Dark Self Cross must be midpoint peg orders, have *RoutingInst* (9303) = `BX`, and have *CrossFlag* (7740) set to either `F` (cross only at the Participant level) or `M` (cross only at the Trading Firm level).

1.8 BATS Market On Close and Trading At Last

BATS Market On Close (MOC) allows Participants to enter orders throughout the trading day that are not executed until the official closing price of the primary exchange is published.

Orders placed into BATS MOC are implicitly hidden but do not need to be Large In Scale (LIS). BATS MOC orders only interact with other BATS MOC orders. Quotes for BATS MOC orders are not represented on any market data feed. Matches in BATS MOC are represented as trades on BATS' market data feeds and may be differentiated. Orders destined for BATS MOC must set *ExecInst* (18) = c or *OrdType* (40) = 5 or B.

During the crossing period, cancellation and modification restrictions apply so Participants must be able to process Execution Report Restatements (*MsgType* (35) = 8 with *ExecType* (150) = D). Participants who never enter MOC orders do not need to process restatements.

BATS MOC functionality is optional and must be enabled at the port level.

Following the publication of the primary exchange's closing price the book moves into the Trading At Last (TAL) phase. Participants can continue entering orders that are immediately executed at the official closing price where liquidity is available.

By default TAL orders will be represented on the BATS market data feed unless the order has been explicitly hidden using *DisplayIndicator* (9479) = I. Hidden orders do not need to be Large In Scale (LIS). The book, if any, is one-sided and consists entirely of imbalance. Orders destined for BATS TAL can use *OrdType* (40) = 1 or 2 or 5 or B.

BATS TAL functionality is enabled by default for all Participants.

For additional information, see the BATS Europe Market On Close Specification.

1.9 Access Fees Returned on Execution Reports

The access fee or rebate associated with each fill is calculated to five decimal places and returned on each execution report. Negative numbers indicate liquidity rebates. Participants should program their systems to read, validate, and pass along this field in order to avoid making software changes to their systems when BATS' fee schedule changes. The sum of the access fees and rebates received during a month should equal access fees charged or rebated on the Participant's monthly bill, rounded appropriately.

Drop copy ports do not accept orders and send execution reports where *ExecType* (150) is Partially Filled (1) or Filled (2). Drop ports for sponsored access may be configured to also send execution reports where *ExecType* (150) is New (0), Canceled (4), or Replaced (5). Drop copy ports may be configured to send various combinations of Participant and clearing identifiers.

1.10 Service Bureau Configuration

Service Bureaus require special configuration. *OnBehalfOfCompID* (115) must be set for Order, Cancel, and Cancel/Replace messages sent to BATS. Orders with an unknown *OnBehalfOfCompID* (115) will be rejected. *ClOrdID* (11) values are required to be unique only within a given *OnBehalfOfCompID* (115). Execution Report and Cancel Reject messages sent by BATS will have the *DeliverToCompID* (128) set. **Orders must be canceled or replaced using the same *OnBehalfOfCompID* (115) as was sent on the original order.**

1.11 Execution Collars

BATS uses market data from the primary exchange for each symbol. This primary best bid and offer (PBBO) is used to create an execution price collar. Executions will not be allowed to occur 20% above the PBBO best ask quote or 20% below the PBBO best bid quote. If an order matches against a resting order, but is outside the 20% price collar, the incoming order will be rejected with reason "price exceeds cross range".

If the primary exchange is not in continuous trading (e.g., is in auction or has closed), the execution collar will be ± 20% of the last regular on-book trade price on the primary exchange. If there has not yet been a trade on the primary exchange today, then no collar is in effect.

1.12 Reserve (Iceberg) Orders

BATS allows the use of *MaxFloor* (111) for entering reserve (iceberg) orders. On reload, BATS' market data feeds show a new OrderID in order to hide the fact that the order is a reserve order.

A port- or firm-level attribute may be enabled which will cause an execution report with *ExecType* (150) = `D` (Restated) to be sent each time an order is reloaded, specifying the new OrderID as will be seen on the market data feeds in *SecondaryOrderID* (198).

The *OrderID* (37) remains constant for the lifetime of the order. Only the *SecondaryOrderID* (198) will change.

Here is a timeline showing an order and its OrderIDs as seen by the Participant and the market data feeds.

#	***OrderID* (37)**	***SecondaryOrderID* (198)**	**Displayed Order ID**	**Notes**
1	`1C3M03000008`	—	`1C3M03000008`	Initial order entry.
2	`1C3M03000008`	`1C3M0300000E`	`1C3M0300000E`	Order reloaded.
3	`1C3M03000008`	`1C3M0300000Q`	`1C3M0300000Q`	Order reloaded again.

1.13 Peg Order Pricing

Pegged orders are priced using the primary best bid and offer (PBBO). If the primary exchange is not in continuous trading (e.g., is in auction or is not currently open), resting pegged orders are canceled back to the Participant and new pegged orders are not allowed to be entered.

1.14 Routing (CYCLE, Parallel-D and Parallel-2D)

Orders marked with an appropriate *RoutingInst* (9303)[4] will be eligible for routing to other market centers if BATS does not offer an equal or better price. Executions for routed orders may be trade reported on the market where the execution occurred and not necessarily on the BATS market data feeds. Notification of each routed execution at another market will be sent to the Participant FIX handler which submitted the routed order.

[4] `R`, `RL`, `RC`, `RND`, `RLD`, `RCD`, `RN2`, `RL2` or `RC2`.

In the event that another market offers a better price, but BATS' connectivity to that market is down and the market in question is otherwise functioning properly, orders which would route to a better quote on that market (but cannot) will be canceled back.

If a market with a better quote is experiencing technical difficulties, BATS may, at its discretion, cease routing to that market and allow orders to post on the BATS book.

Routable orders can route to displayed venues and liquidity partners[5]. BATS will only route to a market if it is in normal, electronic trading. BATS will not route to a primary exchange's auction.

The routing order types and the markets which BATS routes to are subject to change; contact the BATS Europe Trade Desk or your Account Manager for details.

1.15 BATS Plus

Orders marked *RoutingInst* (9303) = `PP or PL`, will be eligible for routing to a single, specified destination.

The resultant trade reporting, market data and FIX excecution reports detail will be the same as for Routable (CYCLE) (1.14). Other market centres will not be considered, even if they offer a better price than BATS or the specified destination.

Some Primary exchanges may not be available with BATS Plus Primary[6]. Contact the BATS Europe Trade Desk or your Account Manager for details.

1.16 Account Field

This field can carry two pieces of information.

Firstly a Central Counterparty (CCP) Account Type prefix.

If this field begins with **H:**, allocate to the house account at the CCP.

If this field begins with **C:**, allocate to the client account at the CCP.

When not prefixed, the *OrderCapacity* (47) is used to determine which CCP account to use. If needed, this can be overridden with explicit configuration by BATS. Non-prefixed or absent accounts would be allocated to *House* account. All CCPs support this feature.

Secondly is a trading account name/number.

With configuration, this can passed to CCPs which support this feature (LCH.Clearnet currently does not). This part must be 16 characters or less. The trading account is configurably available via Drop.

[5] See *ExecInst* (18) = `u, v and w` for details.
[6] *RoutingInst* (9303) = `PP`

2 BATS Europe Specific FIX Fields

The following FIX fields are specific to BATS Europe:

Tag	Name	Description
7740	*CrossFlag*	Enables BATS Dark Self Cross. Force matching only against other orders marked with the same cross flag value. `F` = Match Only at Participant Level `M` = Match Only at Trading Firm Level May only be used with orders marked with *RoutingInst* (9303) = `BX`.
7772	*CentralCounterparty*	Only present on trades. The CCP handling the trade: EMCF = European Multilateral Clearing Facility LCHL = LCH.Clearnet XCLR = SIX x-clear The FIX port must be configured to return this optional field. May not be used in conjunction with *PreventParticipantMatch* (7928).
7928	*PreventParticipant Match*	Participant Trade Prevention: 2 characters (not space separated): 1st character - PTP Modifier: `N` = Cancel Newest `O` = Cancel Oldest `B` = Cancel Both `D` = Decrement Larger[7] / Cancel Smaller `d` = Same as `D` above, but only decrement *LeavesQty* (151). Do not restate *OrderQty* (38). 2nd character - Unique ID Level: `F` = Prevent Match at Participant Level `M` = Prevent Match at Trading Firm Level The Unique ID level (character 2) of both orders must match to prevent a trade. The PTP Modifier (character 1) of the inbound order will be honored, except that if the inbound order specified Decrement and the resting order does not, and the resting order is larger, then both orders will be canceled. This exception is to protect the order entry software for the resting order from receiving an unexpected restatement message. May not be used in conjunction with *CrossFlag* (7740).

[7] Users of PTP modifier `D` must be prepared to receive a FIX Restatement execution report (*ExecType* (150) = `D`) that includes both *OrderQty* (38) and *LeavesQty* (151).

9303	*RoutingInst*	`B` = BATS Only (default) `P` = BATS Only — Post Only (will reject rather than remove visible liquidity) `Q` = BATS Only — Post Only At Limit (remove shares that **improve** upon limit price and up to *MaxRemovePct* (9618) of remaining *OrderQty* (38) **at** limit price) `BD` = BATS Dark Book Only (hidden midpoint peg orders only) `BA` = BATS Automatic Dark Routed (routes to BATS Integrated Book if order is Large In Scale (LIS) or is not a midpoint order, otherwise routes midpoint non-LIS orders to BATS Dark Book) `BX` = BATS Dark Self Cross (crosses against other orders from the same Participant or Trading Firm only; see also *CrossFlag* (7740)) `R` = Parallel-D `RNC` = identical to `R` (CYCLE) `RL` = Parallel-D (reroute on Lock) - allow for use of CYCLE up to limit price on entry and allow reroute via CYCLE after the order has booked *only if another market locks or crosses* the limit `RC` = Parallel-D (reroute on Cross) - similar to `RL` except reroute *only if another market crosses* the limit `RND` = Parallel-D `RLD` = Parallel-D (reroute on Lock) - allow for use of Parallel-D up to limit price on entry and allow reroute via Parallel-D after the order has booked *only if another market locks or crosses* the limit `RCD` = Parallel-D (reroute on Cross) - similar to `RLD` except reroute *only if another market crosses* the limit `RN2` = Parallel-2D `RL2` = Parallel-2D (reroute on Lock) - allow for use of Parallel-2D up to limit price on entry and allow reroute via Parallel-2D after the order has booked *only if another market locks or crosses* the limit `RC2` = Parallel-2D (reroute on Cross) - similar to `RL2` except reroute *only if another market crosses* the limit `PP` = BATS Plus Primary Listing Exchange `PL` = BATS Plus Liquidnet
9479	*DisplayIndicator*	`X` = Displayed Order (default) `I` = Invisible Invisible orders must meet the MiFID CESR requirements for Large in Scale (LIS) unless routed to the BATS Dark Book or BATS Market On Close or BATS Trading At Last.
9617	*ModifySequence*	**Drop only.** Base 36 number of times order has been replaced.
9618	*MaxRemovePct*	For Post Only At Limit (*RoutingInst* (9303) = `Q`), what percentage of the order quantity which remains after price improvement may be removed at the limit price.
9619	*CancelOrigOnReject*	`N` = Leave original order alone (default) `Y` = Cancel original order if replacement fails Default may be configured per port.
9620	*CorrectedPrice*	UCC trade correction message, this holds the corrected price.
9621	*MTFAccessFee*	Only present on trades. Access fee for this fill (up to five decimal places, negative for rebates).
9688	*OrigCompID*	**Drop only.** *TargetCompID* (56) of original FIX execution report. Drop port must be configured to send this optional field.

9689	*OrigSubID*	**Drop only.** *TargetSubID* (57) of original FIX execution report. Drop port must be configured to send this optional field.
9730	*TradeLiquidity Indicator*	Only present on trades. `A` = Added Liquidity `R` = Removed Liquidity `AD` = Added Liquidity for the BATS Dark Pool `RD` = Removed Liquidity from the BATS Dark Pool `AM` = Added Liquidity for BATS Dark Self Cross `RM` = Removed Liquidity for BATS Dark Self Cross `AI` = Added Hidden Liquidity that was price improved `X` = Routed to Another Market To allow for future expansion of this field, please ignore values with an unknown character in the 2nd position.

3 FIX Session Protocol

BATS uses the FIX 4.2 session protocol. The Participant will be provided with a *SenderCompID* (49) and *SenderSubID* (50) that must be sent on every message. The *TargetCompID* (56) for all messages the Participant sends will be `BATS`. The *TargetSubID* (57) is `TEST` for the BATS test system and `PROD` for the BATS production system. All messages the Participant receives will have the sender and target fields swapped, as per the FIX specification.

The following session messages are supported in both directions:

Message	Type	Comment
Logon	`A`	Begin session (or resume a broken session)
Heartbeat	`0`	
Test Request	`1`	
Resend Request	`2`	
Reject	`3`	Malformed message or improper session level handling
Sequence Reset	`4`	Both Gap Fill (*GapFillFlag* (123) = `Y`) and Reset
Logout	`5`	used to gracefully close session

3.1 Sequence Numbers

Sequence numbers, both inbound and outbound, will be reset to one each night during the down time.

Messages are processed in sequence order. Behind sequence messages (other than Sequence Reset — Reset) cause immediate logout. Ahead of sequence messages (other than a Resend Request) trigger a message recovery via a Resend Request.

3.2 Logon

The logon must be the first message sent by the Participant after the TCP connection is established. *EncryptMethod* (98) is ignored (FIX level encryption is not supported).

The IP address of the Participant, the *SenderCompID* (49), *SenderSubID* (50), and *TargetCompID* (56) (`BATS`) and *TargetSubID* (57) (`TEST` or `PROD`) will be validated. If validation fails, the connection will be dropped without a reject (to avoid corrupting the Participant's sequence in the case that the Participant merely mistakenly connected to the wrong port).

If the connection is unexpectedly broken, upon reconnection, the Participant may receive a login reply with a sequence number greater than expected. This means that in-flight messages were missed (likely important execution reports). The Participant should issue a Resend request to retrieve the missed messages.

Similarly, BATS will issue a Resend Request to the Participant for messages that it missed. The Participant may wish to send gap fill messages in place of new orders to avoid submission of potentially stale orders.

HeartBtInt (108) must be specified by the Participant in the Logon message. This value will be clamped between five and 300 seconds and returned in the Logon reply message. We recommend using as low a value as the reliability and latency of your telecommunications channel will allow.

3.3 Heartbeat

A Heartbeat message should be sent if the agreed upon *HeartBtInt* (108) has elapsed since the last message sent. If any message has been sent during the preceding *HeartBtInt* (108), a Heartbeat message need not be sent.

3.4 Test Request

If *HeartBtInt* + 1 seconds have elapsed since the last message received, a Test Request should be issued. If another *HeartBtInt* + 1 seconds go by without receiving a message, the TCP connection should be dropped. This ensures that a broken TCP connection will be detected even if the TCP stack doesn't notice (this has been observed to happen in WAN environments, particularly when a VPN is involved).

3.5 Resend Request

A Resend Request message should be processed even if it is received ahead of sequence. Only after resending the requested range (all market *PossDup* (43) = `Y`), including any gap fills) should Resend Request be issued in the opposite direction.

As discussed in the FIX 4.2 specification, it is possible to send an open or closed sequence range in a Resend Request (an open range uses sequence zero as the *EndSeqNo* (16)). BATS will honor either type of request, but will always issue Resend Requests with a closed sequence range.

3.6 Reject

Session level rejects are used to indicate violations of the session protocol, or missing (or mangled) fields. These are to be expected during development and certification while the Participant's systems are being adapted for BATS, but should be extremely rare in production. Application layer rejects (like Order Reject and Cancel Reject) are normal.

3.7 Sequence Reset

Sequence Reset — Gap Fill messages (*GapFillFlag* (123) = `Y`) must be received in sequence. Any messages (including Gap Fills) sent in response to a Resend Request should have *PossDup* (43) = `Y`.

Sequence Reset — Reset (*GapFillFlag* (123) $\neq$ `Y`) is used only as a last resort, and always by human intervention, to allow an otherwise hopelessly confused session to be resumed. In these cases, all chances at automatic message recovery are lost.

3.8 Logout

Either side may issue a logout to gracefully close the session. The side that issues the logout should process messages normally until it sees the logout reply, and then break the TCP connection. BATS will typically only request logout after the scheduled end of FIX session.

4 Standard FIX Message Header and Trailer

4.1 Header

Tag	Name	Description
8	*BeginString*	`FIX.4.2` Must be the first field in the message.
9	*BodyLength*	Length of message following *BodyLength* field up to and including the delimiter preceding the *CheckSum* (10) field. Must be the second field in the message.
35	*MsgType*	Must be the third field in the message.
34	*MsgSeqNum*	Sequential sequence number for the session.
43	*PossDupFlag*	Indicates a message resent from the admin level (has a duplicate sequence number). Defaults to `N`.
49	*SenderCompID*	ID of sender. Assigned by BATS for messages sent to BATS. (*TargetCompID* (56) for messages from BATS.)
56	*TargetCompID*	ID of destination. `BATS` for messages sent to BATS. (*SenderCompID* (49) for messages from BATS.)
57	*TargetSubID*	Sub ID of destination. `TEST` for messages sent to the BATS test system. `PROD` for messages sent to the BATS production system. (*SenderSubID* (50) for messages from BATS.)
97	*PossResend*	Possible resend flag. BATS has special handling for the *PossResend* for New Order Single messages. See New Order — Single below.
115	*OnBehalfOfCompID*	Service bureau use. Identifies end-client on messages **to** BATS. Must be identifier known to BATS. May be used by non-service bureau to specify which clearing arrangement to use if multiple are configured.
116	*OnBehalfOfSubID*	End-client sub identifier. Four characters, alphanumeric, otherwise not validated. Recorded and returned in *DeliverToSubID* (129). Available via Drop.
122	*OrigSendingTime*	For messages with *PossDupFlag* (43) = `Y`, indicates time that message was first sent.
128	*DeliverToCompID*	Service bureau use. Identifies end-client on message **from** BATS. Must be BATS approved identifier.
129	*DeliverToSubID*	Returns *OnBehalfOfSubID* (116) optionally sent by client.

4.2 Trailer

Tag	Name	Description
10	*CheckSum*	Modulo 256 checksum of all characters in the message up to and including the delimiter preceding the *CheckSum* field. Three digits with leading zeroes if necessary.

5 FIX Application Messages — Participant to BATS

5.1 New Order — Single

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `D`
97	*PossResend*	`N` = indicates a new order (default) `Y` = indicates an application level resend and **is not supported** For reasons of economy, BATS does not track (in primary storage), the *ClOrdID* (11) values of orders that are no longer live. For reasons of performance, BATS does not access secondary storage to enforce unique *ClOrdID* (11) values against orders that are no longer live. Without full duplicate *ClOrdID* (11) value enforcement, it is not possible to safely implement the full behavior specified in the FIX 4.2 protocol for *PossResend* = `Y`. To remain economical, fast, *and* safe, all New Order — Single messages with *PossResend* = `Y` will be simply ignored.
1	*Account*	Optional. Returned on execution reports associated with this order. 16 characters or less (ASCII 33–126). **H:** and **C:** prefix can be used to specify which CCP Account to use. **If configured by BATS**: *House* or *Client* CCP account can be defaulted, regardless of *OrderCapacity* (47). Also the value supplied can be passed to the CCP and made available on the Drop feed.
11	*ClOrdID*	Day-unique ID chosen by client. 20 characters or less. Characters in ASCII range 33–126 are allowed, except for comma, semicolon, and pipe. If the *ClOrdID* matches a live order, it will be rejected as duplicate (unless *PossResend* (97) = `Y`; see above). **Note: BATS only enforces the uniqueness of *ClOrdID* values among currently live orders. However, we *strongly* recommend that you keep your ClOrdID values day unique.**
15	*Currency*	Required if *IDSource* (22) = `4` (ISIN).

18	*ExecInst*	**Single value only (with no trailing space).** `P` = Market Peg (peg buy to PBBO offer, peg sell to PBBO bid) `R` = Primary Peg (peg buy to PBBO bid, peg sell to PBBO offer) `M` = Midpoint (peg to PBBO midpoint) `L` = Alternate Midpoint (less aggressive of midpoint and 1 tick inside PBBO) `c` = BATS Market On Close `u` = BATS + External Dark Only `v` = BATS + External Dark + Lit (default for routable orders) `w` = BATS + External Lit Only **NOTE: Values `L` and `c` differ in meaning from standard FIX 4.2.** All other values are ignored. Midpoint pegged orders (`M` and `L`) and BATS Market On Close orders (`c`) are implicitly hidden (*DisplayIndicator* (9479) = `I`). Only non-displayed Primary and Market Pegs will be accepted at this time.
22	*IDSource*	Values supported by BATS: `2` = SEDOL `4` = ISIN `5` = RIC Required if *Symbol* (55) is not set.
38	*OrderQty*	Number of shares for the order. System-wide limit is 99,999,999 shares.
40	*OrdType*	`1` = Market `2` = Limit `5` = Market On Close `B` = Limit On Close `P` = Pegged Market (`1`) implies *TimeInForce* (59) = `3` (IOC). Pegged requires *ExecInst* (18) = `L`, `M`, `P`, or `R`. Pegged orders may not be routable.
44	*Price*	Limit price.
47	*OrderCapacity*	`A` = Agency `P` = Principal (default) `R` = Riskless
48	*SecurityID*	SEDOL, ISIN, or RIC if *IDSource* (22) is set.
54	*Side*	`1` = Buy `2` = Sell
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
59	*TimeInForce*	`0` = Day `1` = GTC (allowed, but treated as Day) `3` = IOC `6` = GTD (expires at earlier of specified *ExpireTime* (126) or end of day) `8` = Good Through Crossing (expires at the end of the crossing period; can be used for BATS Market On Close orders with *ExecInst* (18) = `c` or *OrdType* (40) = `5` or `B`) **Note: Value `8` is part of standard FIX 5.0SP1.**

60	*TransactTime*	Time order initiated/released. Required by FIX 4.2 but not used by BATS.
110	*MinQty*	**Optional** minimum[8] fill quantity for BATS Only hidden or IOC orders. **Ignored** if order is not BATS Only hidden or IOC. **Default** is zero.
111	*MaxFloor*	Portion of *OrderQty* (38) to display. The balance is reserve. `0` displays entire quantity (default). The displayed quantity of each order at a price level is decremented first. When displayed quantity is fully decremented, it is reloaded up to *MaxFloor* from reserve. May opt-in at the firm or port level to receive a restatement execution report on each reserve reload, allowing a Participant to know the new OrderID as represented on BATS' market data feeds.
126	*ExpireTime*	Required for *TimeInForce* (59) = `6` (GTD) orders, specifies the date and time (in GMT) that the order expires.
207	*SecurityExchange*	Required when *IDSource* (22) = `4` (ISIN).
211	*PegDifference*	**Optional signed** value up to four decimal places[9] is **added** to the result of peg calculation. Default is `0`. Must be ≥ 0 for sell orders. Must be ≤ 0 for buy orders. Must be zero (or not specified) for midpoint peg or non-pegged orders.
439	*ClearingFirm*	Firm that will clear trade. Note: shares storage with *OnBehalfOfCompID* (115). If both fields are set, they must be equal.
440	*ClearingAccount*	Supplemental identifier. Optional. Recorded and returned in execution reports. Available via Drop. Note: shares storage with *OnBehalfOfSubID* (116). If both fields are set, then *OnBehalfOfSubID* (116) takes precedence for Service Bureau connections and *ClearingAccount* takes precedence for other connections.

[8] When removing liquidity, limits the minimum **total** fill size, which may be made up of several **consecutive** smaller fills.
[9] *PegDifference* is rounded (down for buy, up for sell) to fit the tick size.

7928	*PreventParticipant Match*	Participant Trade Prevention: 2 characters (not space separated): 1st character - PTP Modifier: `N` = Cancel Newest `O` = Cancel Oldest `B` = Cancel Both `D` = Decrement Larger[10] / Cancel Smaller `d` = Same as `D` above, but only decrement *LeavesQty* (151). Do not restate *OrderQty* (38). 2nd character - Unique ID Level: `F` = Prevent Match at Participant Level `M` = Prevent Match at Trading Firm Level The Unique ID level (character 2) of both orders must match to prevent a trade. The PTP Modifier (character 1) of the inbound order will be honored, except that if the inbound order specified Decrement and the resting order does not, and the resting order is larger, then both orders will be canceled. This exception is to protect the order entry software for the resting order from receiving an unexpected restatement message. May not be used in conjunction with *CrossFlag* (7740).

[10]Users of PTP modifier `D` must be prepared to receive a FIX Restatement execution report (*ExecType* (150) = `D`) that includes both *OrderQty* (38) and *LeavesQty* (151).

9303	*RoutingInst*	`B` = BATS Only (default) `P` = BATS Only — Post Only (will reject rather than remove visible liquidity) `Q` = BATS Only — Post Only At Limit (remove shares that **improve** upon limit price and up to *MaxRemovePct* (9618) of remaining *OrderQty* (38) **at** limit price) `BD` = BATS Dark Book Only (hidden midpoint peg orders only) `BA` = BATS Automatic Dark Routed (routes to BATS Integrated Book if order is Large In Scale (LIS) or is not a midpoint order, otherwise routes midpoint non-LIS orders to BATS Dark Book) `BX` = BATS Dark Self Cross (crosses against other orders from the same Participant or Trading Firm only; see also *CrossFlag* (7740)) `R` = Parallel-D `RNC` = identical to `R` (CYCLE) `RL` = Parallel-D (reroute on Lock) - allow for use of CYCLE up to limit price on entry and allow reroute via CYCLE after the order has booked *only if another market locks or crosses* the limit `RC` = Parallel-D (reroute on Cross) - similar to `RL` except reroute *only if another market crosses* the limit `RND` = Parallel-D `RLD` = Parallel-D (reroute on Lock) - allow for use of Parallel-D up to limit price on entry and allow reroute via Parallel-D after the order has booked *only if another market locks or crosses* the limit `RCD` = Parallel-D (reroute on Cross) - similar to `RLD` except reroute *only if another market crosses* the limit `RN2` = Parallel-2D `RL2` = Parallel-2D (reroute on Lock) - allow for use of Parallel-2D up to limit price on entry and allow reroute via Parallel-2D after the order has booked *only if another market locks or crosses* the limit `RC2` = Parallel-2D (reroute on Cross) - similar to `RL2` except reroute *only if another market crosses* the limit `PP` = BATS Plus Primary Listing Exchange `PL` = BATS Plus Liquidnet
9479	*DisplayIndicator*	`X` = Displayed Order (default) `I` = Invisible Invisible orders must meet the MiFID CESR requirements for Large in Scale (LIS) unless routed to the BATS Dark Book or BATS Market On Close or BATS Trading At Last.
9618	*MaxRemovePct*	For Post Only At Limit (*RoutingInst* (9303) = `Q`), what percentage of the order quantity which remains after price improvement may be removed at the limit price.
9688	*OrigCompID*	**Drop only.** *TargetCompID* (56) of original FIX execution report. Drop port must be configured to send this optional field.
9689	*OrigSubID*	**Drop only.** *TargetSubID* (57) of original FIX execution report. Drop port must be configured to send this optional field.
	Standard Message Trailer	

5.1.1 Notes on Pegged Orders

Midpoint pegged orders (*ExecInst* (18) = M or L) are implicitly hidden. Midpoint peg orders may execute between the minimum price increment of a stock, except for those stocks which are quoted at a 0.0001

increment. In that case, the peg price will be the less aggressive rounded price (rounded down for buys, rounded up for sells). Midpoint pegs may not use *PegDifference* (211).

Peg orders are prioritised behind non-pegged orders at each price and display level. With regard to hidden peg orders, regular peg orders (*ExecInst* (18) = R or P) have a higher priority than midpoint peg orders ranked at the same price.

Pegged orders will be automatically canceled back to the Participant if BATS loses receipt of market data for any reason or if the primary exchange halts the symbol (including for non-regulatory reasons, such as a volatility interrupt).

5.2 Order Cancel Request

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `F`
97	*PossResend*	`N` = indicates a new cancel (default) `Y` = indicates an application level resend. If *ClOrdID* (11) has not yet been seen, the cancel is treated as normal. If *ClOrdID* (a)lready exists, the resent cancel is ignored.
1	*Account*	Optional. Reflected back on Pending Cancel Execution Report or Cancel Reject associated with this cancel. 16 characters of less (ASCII 33–126). Configurably available via Drop.
11	*ClOrdID*	Day-unique cancel ID chosen by client. 20 characters or less. Characters in ASCII range 33–126 are allowed, except for commma, semicolon, and pipe. Duplicate *ClOrdID*s will be rejected (or ignored if *PossResend* (97) = `Y`.
15	*Currency*	Required if *IDSource* (22) = `4` (ISIN).
22	*IDSource*	Values supported by BATS: `2` = SEDOL `4` = ISIN `5` = RIC Required if *Symbol* (55) is not set.
37	*OrderID*	Order identifier supplied by BATS on the order acknowledgement. (Optional, but recommended for performance.)
38	*OrderQty*	Number of shares for the order. Must match original order.
41	*OrigClOrdID*	*ClOrdID* of the order to cancel.
48	*SecurityID*	SEDOL, ISIN, or RIC if *IDSource* (22) is set.
54	*Side*	`1` = Buy `2` = Sell
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
60	*TransactTime*	Time cancel initiated/released. Required by FIX 4.2 but not used by BATS.
207	*SecurityExchange*	Required when *IDSource* (22) = `4` (ISIN).
	Standard Message Trailer	

5.3 Order Cancel/Replace Request

Only *Price* (44), *OrderQty* (38), and *OrdType* (40) may be adjusted. Any change in *Price* or increase in *OrderQty* will result in the order losing its time priority. *OrdType* may be adjusted from Limit to Market

(but not from Limit to Peg or Peg to Limit).

Other fields (including *ExecInst* (18)) will be ignored, and the value from the original order will be reused.

Changes in *OrderQty* result in an adjustment of the current order's *OrderQty*. The new *OrderQty* does **not** directly replace the current order's *LeavesQty* (151). Rather, a delta is computed from the current *OrderQty* and the replacement *OrderQty*. This delta is then applied to the current *LeavesQty*. If the resulting *LeavesQty* is less than or equal to zero, the order is canceled. This results in safer behavior when the replace request overlaps partial fills for the current order, leaving the Participant in total control of the share exposure of the order.

MaxFloor (111) is preserved from the original order and applied to the new size.

A Cancel/Replace should not be issued until the acknowledgment for the previous Cancel/Replace has been received for that order (or the New Order acknowledgment for the first Cancel/Replace). The FIX handler will reject a new Cancel/Replace if it has not seen the prior Cancel/Replace from the Matching Engine.

Cancel/Replace requests that merely reduce *OrderQty* may be overlapped if the existing *ClOrdID* is reused. This is the only case where reuse of the existing *ClOrdID* is allowed.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `G`
97	*PossResend*	`N` = indicates a new cancel/replace (default) `Y` = indicates an application level resend. If the *ClOrdID* does not indicate an already pending cancel/replace, the cancel/replace is treated as normal. If *ClOrdID* does indicate an already pending cancel/replace, then the resent cancel/replace is ignored.
1	*Account*	Optional. Returned on execution reports associated with this order. 16 characters or less (ASCII 33–126). **H:** and **C:** prefix can be used to specify which CCP Account to use. **If configured by BATS**: *House* or *Client* CCP account can be defaulted, regardless of *OrderCapacity* (47). Also the value supplied can be passed to the CCP and made available on the Drop feed.
11	*ClOrdID*	Day-unique ID chosen by client. 20 characters or less. Characters in ASCII range 33–126 are allowed, except for comma, semicolon, and pipe. If the *ClOrdID* matches a live order, it will be rejected as duplicate (unless *PossResend* (97) = `Y`; see above). **Note: BATS only enforces the uniqueness of *ClOrdID* values among currently live orders. However, we *strongly* recommend that you keep your ClOrdID values day unique.**
15	*Currency*	Required if *IDSource* (22) = `4` (ISIN).
22	*IDSource*	Values supported by BATS: `2` = SEDOL `4` = ISIN `5` = RIC Required if *Symbol* (55) is not set.
37	*OrderID*	Order identifier supplied by BATS on the order acknowledgement. In the case of multiple changes to a single order, this should be the *OrderID* from the most recent acknowledgement.

38	*OrderQty*	Number of shares for the order. This will modify the *OrderQty* of the current order, it does **not** directly set the remaining quantity.
40	*OrdType*	`1` = Market `2` = Limit `5` = Market On Close `B` = Limit On Close `P` = Pegged Market (`1`) implies *TimeInForce* (59) = `3` (IOC). Pegged requires *ExecInst* (18) = `L`, `M`, `P`, or `R`. Pegged orders may not be routable.
41	*OrigClOrdID*	*ClOrdID* of the order to replace. In the case of multiple changes to a single order, this will be the *ClOrdID* of the most recently accepted change.
44	*Price*	Limit price.
48	*SecurityID*	SEDOL, ISIN, or RIC if *IDSource* (22) is set.
54	*Side*	`1` = Buy `2` = Sell
55	*Symbol*	Security symbol. See Symbology (p. 4) for additional notes.
60	*TransactTime*	Time cancel/replace initiated/released. Required by FIX 4.2 but not used by BATS.
207	*SecurityExchange*	Required when *IDSource* (22) = `4` (ISIN).
9619	*CancelOrigOnReject*	`N` = Leave original order alone (default) `Y` = Cancel original order if replacement fails Default may be configured per port.
	Standard Message Trailer	

6 FIX Application Messages — BATS to Participant

6.1 Execution Report

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `8`
1	*Account*	Copied from order, if present. (Not available via Drop.)
6	*AvgPx*	Average fill price.
11	*ClOrdID*	*ClOrdID* of the order being accepted, executed, or rejected. –or– *ClOrdID* of the cancel or replace **request**. –or– *ClOrdID* of the order subject to unsolicited cancel (*OrigClOrdID* (41) will not be present).
14	*CumQty*	Cumulative quantity of shares executed for this order.
15	*Currency*	Copied from order, if present.
17	*ExecID*	Day-unique ID of execution message.
18	*ExecInst*	Copied from order, if present.
20	*ExecTransType*	`0` = New
22	*IDSource*	Copied from order, if present.
31	*LastPx*	Price of this fill (zero for non-fills).
32	*LastShares*	Quantity of shares traded on this fill (zero for non-fills).
37	*OrderID*	Order identifier supplied by BATS.
38	*OrderQty*	Copied from order.
39	*OrdStatus*	State of order. `0` = New `1` = Partially Filled `2` = Filled `4` = Canceled `5` = Replaced `6` = Pending Cancel `8` = Rejected `A` = Pending Ack `E` = Pending Replace
41	*OrigClOrdID*	*ClOrdID* of the order being canceled or replaced (for a solicited cancel or cancel/replace, otherwise not present).
44	*Price*	Copied from order.
48	*SecurityID*	Copied from order, if present.
52	*SendingTime*	GMT date and time that message was sent.
54	*Side*	Copied from order.
55	*Symbol*	Copied from order, if present.

58	*Text*	If present, indicates reason for the message. Format is one letter reason code followed by colon and space followed by free form text message. Reason codes are: `A` = Admin `D` = Duplicate *ClOrdID* `H` = Halted `K` = Order Rate Threshold Exceeded `L` = Price Exceeds Cross Range `N` = Ran Out of Liquidity to Execute Against `O` = *ClOrdID* Doesn't Match a Known Order `P` = Can't Modify an Order That is Pending Fill `Q` = Waiting For First Trade `R` = Routing Unavailable `U` = User Requested `V` = Would Wash `W` = Add Liquidity Only Order Would Remove `X` = Order Expired `Y` = Symbol Not Supported `Z` = Unforseen Reason `r` = Reserve Reload
59	*TimeInForce*	Copied from order.
60	*TransactTime*	Time transaction occurred.
103	*OrdRejReason*	Optionally set when *ExecType* (150) = `8` (Rejected). `0` = Broker Option `1` = Unknown Symbol `2` = Exchange Closed `3` = Order Exceeds Limit `5` = Unknown Order `6` = Duplicate Order `8` = Stale Order
111	*MaxFloor*	Copied from order.
126	*ExpireTime*	Copied from order if *TimeInForce* (59) = `6` (GTD).
150	*ExecType*	Reason for this execution report. `0` = New (acknowledgement of new order) `1` = Partial Fill `2` = Fill `4` = Canceled `5` = Replaced `8` = Rejected `D` = Restated
151	*LeavesQty*	Quantity of shares still open for further execution. Will be zero if order is dead, otherwise will be *OrderQty* − *CumQty*. **Note**: It is possible for *LeavesQty* to be zero when *ExecType* (150) = `5` indicating that the order is dead.
198	*SecondaryOrderID*	Must request opt-in at firm or port level to receive this field. Present on Prevent Participant Match triggered cancel/restatement. Denotes the BATS *OrderID* (37) of contra side of prevented match. Present on a restatement execution report for reload of a reserve (iceberg) order. Denotes the new BATS OrderID which will be present on BATS' market data feeds.
207	*SecurityExchange*	Copied from order, if present.

375	*ContraBroker*	Only present on trades. `BATS`: execution on BATS Integrated or Dark Pool `LP`: execution on order routed to external liquidity provider or the ISO Market Identification Code (MIC)[11] for executions on orders routed to another market.
378	*ExecRestatement Reason*	Required when *ExecType* (150) = `D` (Restated). `4` = Broker option; optionally sent during reload of a reserve (iceberg) order `5` = Partial decline of *OrderQty*
382	*NoContraBrokers*	Only present on trades. Always `1`.
439	*ClearingFirm*	Copied from order, if present.
440	*ClearingAccount*	Copied from order, if present.
7772	*CentralCounterparty*	Only present on trades. The CCP handling the trade: EMCF = European Multilateral Clearing Facility LCHL = LCH.Clearnet XCLR = SIX x-clear The FIX port must be configured to return this optional field. May not be used in conjunction with *PreventParticipantMatch* (7928).
9621	*MTFAccessFee*	Only present on trades. Access fee for this fill (up to five decimal places, negative for rebates).
9730	*TradeLiquidity Indicator*	Only present on trades. `A` = Added Liquidity `R` = Removed Liquidity `AD` = Added Liquidity for the BATS Dark Pool `RD` = Removed Liquidity from the BATS Dark Pool `AM` = Added Liquidity for BATS Dark Self Cross `RM` = Removed Liquidity for BATS Dark Self Cross `AI` = Added Hidden Liquidity that was price improved `X` = Routed to Another Market To allow for future expansion of this field, please ignore values with an unknown character in the 2nd position.
	Standard Message Trailer	

6.2 Cancel Reject

Rejects a Cancel or Cancel/Replace request.

When a Cancel/Replace is rejected, by default, the original order is left alive. A Cancel Reject should not be used as a sign that the original order has been canceled. Even if the *CancelOrigOnReject* (9619) = `Y` option is being used, a separate "unsolicited" cancel will be sent to close out the original order.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `9`
1	*Account*	Copied from Cancel or Cancel/Replace request.
11	*ClOrdID*	*ClOrdID* from the Cancel or Cancel/Replace request.
37	*OrderID*	*OrderID* of the order that failed to be canceled or replaced. `NONE` if *CxlRejReason* (102) = `1` (Unknown Order).

[11] ISO 10383, see http://www.iso15022.org/MIC/homepageMIC.htm for details

39	*OrdStatus*	State of order that failed to be canceled or replaced.
41	*OrigClOrdID*	*ClOrdID* of the order that failed to be canceled or replaced.
58	*Text*	Free-form text message.
102	*CxlRejReason*	`0` = Too Late to Cancel `1` = Unknown Order `3` = Already Pending Cancel or Pending Replace
434	*CxlRejResponseTo*	`1` = Cancel `2` = Cancel/Replace
	Standard Message Trailer	

6.3 Trade Cancel/Correct

Trade Cancel/Correct (UCC) is an optional message that must be enabled at the port level. It may be enabled for current-day only or for all cancels and corrections. Only the price of a trade may be corrected, all other details remain the same. Trade cancels and corrections do not alter live order state.

Tag	Name	Description
	Standard Message Header	*MsgType* (35) = `UCC`
11	*ClOrdID*	*ClOrdID* of the order whose trade is being canceled or corrected.
15	*Currency*	Copied from order if *IDSource* (22) = `4` (ISIN).
17	*ExecID*	Day-unique ID of execution message.
19	*ExecRefID*	Refers to the *ExecID* (17) of the execution being canceled or corrected.
20	*ExecTransType*	`1` = Cancel `2` = Correct
22	*IDSource*	Copied from order being canceled or corrected, if present.
31	*LastPx*	Price on the original trade being canceled or corrected.
32	*LastShares*	Quantity of shares on the original trade being canceled or corrected.
37	*OrderID*	*OrderID* of the order whose trade is being canceled or corrected.
42	*OrigTime*	Date and time of the original trade, in GMT.
48	*SecurityID*	Copied from original order being canceled or corrected if *IDSource* (22) = `2` (SEDOL), `4` (ISIN), or `5` (RIC) was used.
54	*Side*	Copied from trade being canceled or corrected.
55	*Symbol*	Copied from original order being canceled or corrected.
60	*TransactTime*	Date and time of the cancel or correction.
207	*SecurityExchange*	Copied from order being canceled or corrected if *IDSource* (22) = `4` (ISIN) was used.
439	*ClearingFirm*	Copied from trade being canceled or corrected, if present.
440	*ClearingAccount*	Copied from trade being canceled or corrected, if present.
9620	*CorrectedPrice*	The corrected price of the trade. Only set if *ExecTransType* (20) = `2` (Trade Correct).
9730	*TradeLiquidity Indicator*	Copied from trade being canceled or corrected.
	Standard Message Trailer	

7 Common Session Level Issues

BATS uses FIX 4.2 as specified by the FPL Document Version 4.2 (with Errata 20010501) with business level extensions as described in this document. The session level of the FPL specification is followed as closely as possible.

The version with errata cleared up many session level ambiguities present in the earlier version 4.2 (March 1, 2000). The following sections emphasize a few common problem areas in implementations of the FIX session protocol.

Typographical conventions:

- Anchor locations in the FPL document are shown in blue.
- Text in **bold** was emphasized in the original FPL specification.
- Emphasis added by BATS is shown in purple.
- Notes added by BATS are shown in green.

7.1 Ordered Message Processing

From Financial Information Exchange Protocol/FIX Message Format and Delivery/Ordered Message Processing:

> The FIX protocol assumes complete ordered delivery of messages between parties. Implementers should consider this when designing message gap fill processes. Two options exist for dealing with gaps, either request all messages subsequent to the last message received or ask for the specific message missed while maintaining an ordered list of all newer messages. For example, if the receiver misses the second of five messages, the application could ignore messages 3 through 5 and generate a resend request for messages 2 through 5, **or**, preferably 2 through 0 (where 0 represents infinity). Another option would involve saving messages 3 through 5 and resending only message 2. In both cases, messages 3 through 5 should not be processing before message 2.

7.2 Logon

From Financial Information Exchange Protocol/Session Protocol/Logon:

> After the initiator has been authenticated, the acceptor will respond immediately with a confirming *Logon* message.

7.3 Message Recovery

From Financial Information Exchange Protocol/Session Protocol/Message Recovery:

> When the incoming sequence number does not match the expected number, corrective processing is required. Note that the SeqReset-**Reset** message ([BATS: this refers only to *GapFillFlag* (123) = `N`] used only to recover from a disaster scenario vs. normal resent request processing) is an exception to this rule as it should be processed without regards to its *MsgSeqNum* (34). **If the incoming message has a sequence number less than expected and the *PossDupFlag* (43) is not set, it indicates a serious error. It is strongly recommended that the session be terminated and manual intervention be initiated**. If the incoming sequence number is greater than expected, it indicates that messages were missed and retransmission of the messages is requested via the *Resend Request* (see earlier section, *Ordered Message Processing*).
>
> ...

If there are consecutive administrative messages to be resent, it is suggested that only one *SeqReset-GapFill* message be sent in their place. The sequence number of the *SeqReset-GapFill* message is the next expected outbound sequence number. The *NewSeqNo* (36) field of the GapFill message contains the sequence number of the highest administrative message in the group plus 1. For example, during a Resend operation there are 7 sequential administrative messages waiting to be resent. They start with sequence number 9 and end with sequence number 15. Instead of transmitting 7 GapFill messages (which is perfectly legal, but not network friendly), a *SeqReset-GapFill* message may be sent. **The sequence number of the Gap Fill message is set to 9 because the remote side is expecting that as the next sequence number.** The *NewSeqNo* (36) field of the Gap Fill message contains the number 16, because that will be the sequence number of the next message to be transmitted.

Sequence number checking is a vital part of FIX session management. However, a discrepancy in the sequence number stream is handled differently for certain classes of FIX messages. The table below lists the actions to be taken when the incoming sequence number is greater than the expected incoming sequence number.

> **NOTE: In all cases except the Sequence Reset – Reset message, the FIX session should be terminated if the incoming sequence number is less than expected and the *PossDupFlag* (43) is not set. A *Logout* message with some descriptive text should be sent to the other side before closing the session.**

Response by Message Type

Message Type	Action to Be Taken on Sequence # Mismatch
Logon	Must always be the first message transmitted. Authenticate and accept the connection. After sending a *Logon* confirmation back, send a *ResendRequest* if a message gap was detected in the *Logon* sequence number.

...

7.4 Resend Request

From Financial Information Exchange Protocol/Administrative Messages/Resend Request:

> Note: the sending application may wish to consider the message type when resending messages; e.g., if a new order is in the resend series and a significant time period has elapsed since its original inception, the sender may not wish to retransmit the order given the potential for changed market conditions. (The Sequence Reset-Gap Fill message is used to skip message that a sender does not wish to resend.)

7.5 Sequence Reset – Gap Fill

From Financial Information Exchange Protocol/Administrative Messages/Sequence Reset (Gap Fill):

> The sequence reset message is used by the sending application to reset the incoming sequence number on the opposing side. This message has two modes: "Sequence Reset – Gap Fill when *GapFillFlag* (123) is 'Y' and "Sequence Reset – Reset" when *GapFillFlag* (123) is 'N' or not present. The "Sequence Reset – Reset" mode should **only** be used to recover from a disaster situation which cannot e otherwise recovered via "Gap Fill" mode. The sequence reset message can be used in the following situations:

- During normal resend processing, the sending application may choose not to send a message (e.g., an aged order). The Sequence Reset – Gap Fill is used to mark the place of that message.
- During normal resend processing, a number of administrative messages are not resent, the Sequence Reset – Gap Fill message is used to fill the sequence gap created.

. . .

The sending application will initiate the sequence reset. **The message in all situations specifies the *NewSeqNo* (36) to reset as the value of the next sequence number immediately following the messages and/or sequence numbers being skipped.**

. . .

If the *GapFillFlag* (123) field is present (and equal to 'Y'), the *MsgSeqNum* (34) should conform to standard message sequencing rules (i.e., the *MsgSeqNum* (34) of the SeqReset-GapFill message should represent the beginning *MsgSeqNum* (34) in the gap fill range because the remote side is expecting that next message).

The sequence reset can only increase the sequence number. If a sequence reset is received attempting to decrease the next expected sequence number, the message should be rejected and treated as a serious error. It is possible to have multiple resend requests issued in a row (i.e., 5 to 10 followed by 5 to 11). If sequence number 8, 10, and 11 represent application messages while 5–7 and 9 represent administrative messages, the series of messages as a result of the resend request may appear as SeqReset-GapFill with *NewSeqNo* (36) of 8, message 8, SeqReset-GapFill with *NewSeqNo* (36) of 10, and message 10. This could then be followed by SeqReset-GapFill with *NewSeqNo* (36) of 8, message 8, SeqReset-GapFill with *NewSeqNo* (36) of 10, message 10, and message 11. One must be careful to ignore the duplicate SeqReset-GapFill which is attempting to lower the next expected sequence number. This can be detected by checking to see if its *MsgSeqNum* (34) is less than expected. If so, the SeqReset-GapFill is a duplicate and should be discarded.

8 FIX Drop

BATS offers two types of FIX Drop ports (Standard FIX Drop and Order by Order FIX Drop). Both port types do not accept orders. Their purpose is to provide real time information about order flow. They may be configured to send order flow based on various combinations of information relating to specific Participants, trading firm identifiers, and/or sessions. With proper authorisation (e.g., clearing or sponsored relationships), a single FIX Drop session can be used to obtain information about multiple Participants.

8.1 Standard FIX Drop

Standard FIX drop ports only send execution information on fills (i.e., execution reports where *ExecType* (150) = `1` (Partially Filled) or `2` (Filled).

8.2 Order By Order FIX Drop

All order message types are supported including, but not limited to:

- *ExecType* (150) = `0`: Acknowledgments
- *ExecType* (150) = `1` or `2`: Partially Filled, Filled
- *ExecType* (150) = `4`: Canceled
- *ExecType* (150) = `5`: Replaced
- *ExecType* (150) = `8`: Rejected
- *MsgType* (35) = `9`: Order Cancel Reject
- *MsgType* (35) = `UCC`: Trade Cancel/Correction (optionally, if configured at the port level)

If rejects or cancels are due to incomplete or incorrect clearing information, they may be unavailable on Order by Order FIX Drop ports.

8.3 Port Options

Both types of FIX Drop ports can be optionally configured with the following features:

1. Choice of various BATS supported symbology types.
2. Sending of Trade Cancel/Correction (*MsgType* (35) = `UCC`) messages. Please note that enabling these messages will be dependent on enabling of trade cancels/corrections on the corresponding FIX order entry ports.
3. Enable unique wash execution identifiers.

9 FIX Differences Between US and Europe

This section describes, in detail, the differences between the FIX implementations of the BATS Exchange in the United States and BATS MTF in Europe. The FIX session level implementation and supported messages are identical between the Exchange and the MTF.

Tick Sizes

In the United States, there is currently a single tick band for all symbols. Prices less than $1.00 have a tick size of $0.0001. Prices greater than $1.00 have a tick size of $0.01. On the MTF, tick sizes vary by market, price, and symbol. Reference data files are available daily which enumerate the tick sizes.

Routing Instructions

The values which are common across the Exchange and MTF are *RoutingInst* = `B`, `P`, `R`, `Q`.

Display Indicator

Values for the *DisplayIndicator* (9479) are different. The MTF only defines *DisplayIndicator* (9479) = `X` (visible) and `I` (invisible). BATS Exchange offers additional values for price sliding.

Bypass Hidden

The *BypassHidden* (9687) is not supported on the MTF.

Trade Liquidity Indicator

Values for the *TradeLiquidityIndicator* (9730) are different. Values which are common across the Exchange and the MTF are *TradeLiquidityIndicator* (9730) = `A` (Added) and `R` (Removed). The Exchange offers values for routing which the MTF does not. The MTF offers values for fills done in the BATS Dark Pool which the Exchange does not.

Discretionary Orders

For regulatory reasons, discretionary orders, specified on the Exchange with *DiscretionAmount* (9622), are not supported on the MTF.

Sell Short

The sell short *Side* (54) = `5` value is not supported on the MTF.

Execution Instruction

The Exchange supports *ExecInst* (18) values for intermarket sweep orders (`18=f`) and Dark Scan (`18=z`) which are not supported on the MTF. The MTF supports `18=u` which is not supported on the Exchange.

Working and Display Price Fields

The *WorkingPrice* (9690) and *InitialDisplayPrice* (9691) fields are not supported on the MTF. These fields are relevant for price sliding which is only supported on the Exchange.

Market on Close, Trading at Last

MOC orders are not supported on the Exchange. MOC orders use *ExecInst* (18) = `c`, may use *TimeInForce* (59) = `8` and have *OrdType* (40) = `B` or `5`.

Symbology

The MTF allows specifying SEDOL, ISIN, RIC, or Common symbologies which require the use of *SecurityExchange* (207), *Currency* (15), *SecurityID* (48), *IDSource* (22), and/or *Symbol* (55). The Exchange uses *Symbol* (55) and *SymbolSfx* (65) only.

Market Hours

Pre- and post-market trading are supported on the Exchange, but not on the MTF.

Central Counterparty

The optional reply field *CentralCounterparty* (7772) is only supported on the MTF.

Contra Broker

The *ContraBroker* (375) tag will contain completely different values on the MTF or the Exchange.

10 Support

Please email questions or comments regarding this specification to `tradedeskeurope@batstrading.com`.

11 Revision History

9 July 2008	Initial draft version.
16 July 2008	Europe URL and email address.
2 September 2008	Updated tick sizes. Removed references to *WorkingPrice* (9690). Removed wording that prices will be slid to a less aggressive amount if they do not fit the tick size for a symbol. Orders will be rejected if they do not fit the tick size. Added *DisplayIndicator* (9479) to "BATS Specific Fields" and to allowed fields for "New Order – Single". Added section entitled "Hidden Orders".
8 September 2008	Added information about the *Text* (58) that will be received in the event of an order which, if executed, would happen outside the allowed price collar. Added clarification that *OrdRejReason* (103) will not always be sent on order rejects.
1 October 2008	Adjusted price collar for allowed executions to be up to 20% away from the market.
11 November 2008	Added ability to allocate trades to house or client account regardless of order capacity by using the *Account* (1) field. Updated front page text to note FSA authorisation.
16 December 2008	Added information on how notional value is determined for different order types.
3 February 2009	Added details for new Trade Cancel/Correct (UCC) message type.
26 March 2009	Discretionary orders are no longer permitted.
27 March 2009	Added new order type Post Only At Limit (*RoutingInst* (9303) = `Q`) and new tag *MaxRemovePct* (9618).
9 July 2009	Added BATS Dark Pool routing. New values for *RoutingInst* (9303) and *TradeLiquidityIndicator* (9730).
15 July 2009	Added ability to post only to BATS Dark Pool.
16 July 2009	Added BATS Market On Close. New values for *ExecInst* (18), *OrdType* (40), *TimeInForce* (59), and *ExecType* (150).
23 July 2009	Added Participant Trade Prevention functionality. New tag of *PreventParticipantMatch* (7928). Added new section "FIX Differences Between US And Europe". Documented *OrdStatus* (39) = `A`. Only occurs on a Cancel/Replace reject if the FIX Handler is awaiting an acknowledgement on a new order from the Matching Engine.
30 July 2009	*RoutingInst* (9303) = `BA` or `PA` will now route any order which is not eligible for the BATS Dark Pool to the integrated book.
24 August 2009	Updates to participant match prevention.
9 September 2009	Removed reference to *WorkingPrice* (9690) which isn't used on BATS Europe. Fixed wording about which symbology tags are required on cancel order messages.
7 October 2009	Central Counterparty updates.
19 October 2009	Clarified in the Execution Report that *TradeLiquidityIndicator* (9730) can also take on values `AD` and `RD` for executions which occur in the BATS Dark Pool.

23 October 2009	Noted that *ExecInst* (18) will be returned on Execution Reports if set on the original order.
19 November 2009	Version 2.0. Added new "FIX Drop" section. Numerous formatting changes. *ExecType* (150) = `3` (Done for Day) was listed as a possible value BATS would send, but it is not.
24 January 2010	Version 2.1. Removed *RoutingInst* (9303) = `PA` or PD. Added dark and onward routing. New values for *RoutingInst* (9303), *ExecInst* (18), *TradeLiquidityIndicator* (9730), *Text* (58) (reject message if routing is unavailable).
24 January 2010	Version 2.2. New values for *ContraBroker* (375).
18 February 2010	Version 2.3. Added Common Session Level Issues (§ 7, p. 28). Removed section about Market BBO and added Execution Collars (§ 1.11, p. 7) and Peg Order Pricing (1.13).
9 March 2010	Version 2.4. *RoutingInst* (9303) = `RL` or `RC` will re-route (RECYCLE) a booked order if another market locks or crosses the limit.
16 March 2010	Version 2.5. By default BATS Trading At Last orders are now visible unless explicitly hidden using *DisplayIndicator* (9479) = `I`. *OrdType* (40) values `1` (Market) and `2` (Limit) are now accepted during the TAL phase.
15 April 2010	Version 2.6 Removed order quantity limit of 999,999. The new limit is 99,999,999 shares, but may be lowered if requested by a Participant or Sponsor.
30 April 2010	Version 2.7 Added BATS Dark Self Cross (§ 1.7, p. 6). Added *CrossFlag* (7740). Added new value for *RoutingInst* (9303) (`BX` = BATS Dark Self Cross). New values for *TradeLiquidityIndicator* (9730) (`AM`, `RM`). New values for *ContraBroker* (375).
7 May 2010	Version 2.8 Added *PreventParticipantMatch* (7928) = `d`.
20 May 2010	Clarified that Post Only orders will only reject when removing visible liquidity.
1 June 2010	Version 2.9 Added BATS Plus (§ 1.15, p. 8) and Account Field (§ 1.16, p. 8) sections. Update (§ 1.14, p. 7) to clarify differences to BATS Plus. New values for *RoutingInst* (9309) and clarification of *Account* (1)
26 August 2010	Version 2.10 Fixed minor typographical errors.
27 August 2010	Version 2.11 Parallel-D routing strategy. New values for *RoutingInst* (9303).
12 October 2010	Version 2.12 Restatement execution reports may be optionally received on reserve reload, allowing Participants to know the new OrderID that will be shown on BATS' market data feeds. New initial character on *Text* (58) field of `r` indicating the restatement was sent due to reserve reload. See new section § 1.12, p. 7.
22 October 2010	Version 2.13 Midpoint peg orders may now have a limit price at one-half the tick size (§ 1.4, p. 5).

25 November 2010	Version 2.14 Parallel-2D routing strategy. New values for *RoutingInst* (9303). Added *Side* (54) to Trade Cancel/Correct (UCC) message type.
14 January 2011	Version 2.15 Noted that midpoint peg orders at half-tick sizes may only specify limit prices out to a maximum of four decimal places.
2 February 2011	Version 2.16 *OrdStatus* (39) = `C` was documented, but never sent in practice. Removed this value from the specification.
8 February 2011	Version 2.17 Noted that midpoint peg orders are not tick size validated.
22 March 2011	Version 2.18 Corrected various instances where MBBO was incorrectly referenced instead of PBBO.
5 April 2011	Version 2.19 Removed `1` and `2` from values which are communicated in *ExecTransType* (20) for execution reports. Removed *ExecRefID* (19) from execution reports as it is never sent. *AvgPx* (6) was missing from the execution report documentation.
14 April 2011	Version 2.20 Updated *RoutingInst* (9303) values of `R`, `RL`, and `RC` to note that Parallel-D is used, not CYCLE.
27 April 2011	Version 2.21 Added new value `AI` for *TradeLiquidityIndicator* (9730).