



Aquis Trading Protocol (ATP) Specification

15th August 2017
Version 2.0 (MiFID II)

Table of Contents

Version History.....	iii
1 Introduction.....	1
1.1 Connectivity.....	1
1.2 Enquiries / Support.....	1
2 Overview	2
2.1 Message Structure and Sequence Number Logic	2
2.2 Order Reference	2
2.3 Clearing Configuration	2
2.4 Market at Close	3
2.5 Cancel on Disconnect	3
2.6 Self-Trade Prevention	4
2.7 Post-Only Order Type	4
3 ATP Message Formats	5
3.1 Data Types	5
3.2 Message Structure	5
3.2.1 Message Header	5
3.3 Session Level Messages	6
3.3.1 Login Message	6
3.3.2 Login Response Message.....	6
3.3.3 Heartbeat.....	7
3.3.4 Logout Request Message	7
3.3.5 Logout Message.....	8
3.4 Business Messages	8
3.4.1 Order Add Message	8
3.4.2 Order Cancel Message	9
3.4.3 Order Modify Message	10
3.4.4 Order Add Response Message	11
3.4.5 Order Cancel Response Message	12
3.4.6 Order Modify Response Message.....	12
3.4.7 Trade Message	12
3.4.8 Trade Bust Message	13
4 Status and Reason Codes	13
4.1 Order Status	13
4.2 Modification Codes.....	13
4.3 Cancel Reason.....	14
4.4 Reject Reasons	14

Version History

Version	Date	Comments
1.0	27 Jun 2013	Initial version
1.1	21 Oct 2013	Include order status and reject codes; protocol version is 1.1; update Trade message structure; describe message replay behaviour; clarifications and typo corrections.
1.2	20 Jan 2014	Update to message replay behaviour after disconnect; include timestamp on business messages to member; include securityId on Trade message; addition of userTag field.
1.3	14 July 2014	Addition of MaC order type; return cancel reason on Order Cancel Response message.
1.3.1	23 Sep 2014	Support Riskless Principal order capacity; Update reject reason codes.
1.3.2	10 Mar 2015	New cancel reason code for self-trade prevention; update to MaC behaviour; New liquidity indicator codes.
1.3.3	10 Feb 2016	Support for Post Only order type.
1.4	17 Mar 2017	Addition of modification codes (for ATP version 1.4 and above).
2.0	15 Aug 2017	MiFID II fields included and updated.

1 Introduction

Aquis Exchange (Aquis) has developed a proprietary binary protocol for efficient, streamlined and low-latency order entry and trading activity – Aquis Trading Protocol (ATP).

This document describes the protocol and message formats for this trading interface. It is intended for those firms which are planning to develop against the protocol.

1.1 Connectivity

Aquis will provide trading Members that wish to use ATP to access the platform with the necessary IP address and port information to establish a TCP/IP connection for each of their trading sessions.

In addition, a senderId and password for each session will be agreed.

One or more connections will be provided to the customer test environment and, on successful certification, connection details to the primary and backup production trading environments will be supplied. Connectivity options should be discussed with the Aquis networks team.

1.2 Enquiries / Support

Please contact the Aquis support team (email support@aquis.eu) for any questions related to this document.

2 Overview

2.1 Message Structure and Sequence Number Logic

The message structure principles for ATP are to provide efficient, fixed-length messages with binary field data directly aligned to the internal message structure used by the trading system.

To avoid unnecessary traffic on the internal trading system, session messages (login, logout and heartbeat) are not sequenced – only business messages relating to orders and trades are sequenced and recoverable. Sequenced messages belong to one of two streams - trading Member to Aquis or Aquis to trading Member - each with their own numbering starting at 1 each trading day.

The sequence number of business messages to Aquis must always increase. A message with a sequence number that is lower than, or the same as, one already seen will lead to the session being terminated with an appropriate error code. The sequence number of business messages sent from Aquis to the trading Member will also increase, incremented by 1 on each business message during the day.

2.2 Order Reference

In the ATP protocol it is not necessary for a separate client order identifier to be sent on an Order Add message. The sequence number of the Order Add message is used by both sides as an 'order reference' to identify to that order in subsequent messages.

For example, if an Order Add message is sent in with *msgSeqNo* = 121, then any subsequent Order Modify messages for that order should carry *orderRef* = 121. Similarly an Order Cancel message should also carry the *orderRef* value. Responses and Trade messages from Aquis will also carry *orderRef* = 121 to refer to the relevant order.

Note that if an Order Modify or Order Cancel message is sent in with, for example, *msgSeqNo* = 143 then the Order Modify Response or Order Cancel Response message would carry *orderRef* = 121 and *requestRef* = 143 to identify both the order and the request.

For the convenience of trading Members a *userTag* field has also been made available on Order Add, Order Cancel and Order Modify messages. This value is echoed back by Aquis on the related Response messages and also on Trade messages. This allows the trading Member to link business messages back to their own order tracking system using their own internal identifiers. Note that the *userTag* is a free-format field and is not validated by Aquis; it may be left blank if not used by the trading Member.

2.3 Clearing Configuration

Before trading is permitted in any particular market, the trading Member must confirm that they have clearing arrangements in place with a nominated CCP. These choices are verified and set up by Aquis as part of the Member's trading configuration.

The Member may communicate on a per-order basis whether they wish for trades on a particular order to be allocated to the house account at the CCP or to a specific client account. Client account details should be established with the CCP involved and verified with Aquis.

The Member may also register for 'self-trade suppression' whereby if a trading Member trades with itself then the trade is not submitted to a CCP for clearing. Instead, it is marked as a self-trade to be dealt with by the Member internally.

2.4 Market at Close

Trading Members wishing to enter orders to the Aquis Market at Close (MaC) should contact Aquis support to be certified for this service.

MaC orders may be entered during the continuous trading phase or when the Aquis MaC opens, immediately after the continuous trading phase closes. Orders submitted during continuous trading are held and then are included and published in the MaC when it opens.

Throughout the continuous trading phase and the initial unlocked MaC phase, orders may be entered, modified and cancelled.

Should a particular security at the Market of Listing not enter an auction phase within 30 seconds of the close of continuous trading on Aquis, orders in that security will be cancelled back to the Member.

At MaC lock time, unpriced Trade messages are sent for the matched quantity. If there is residual unmatched order quantity, or the order has not matched at all, then a cancel is sent. The cancel reason in the message identifies that this a cancellation of the unmatched quantity; the trading Member should expect to receive priced Trade messages for matched orders when the firm MaC closing price is established.

If for any reason the Market of Listing auction is cancelled or extended for a security, all orders in that security will be cancelled back to the Member. The cancel reason in the message identifies this as a MaC cancel due to a problem. If this is sent after lock time, it means that the MaC is aborted and any indicative trades will not be priced – the full order quantity is cancelled back to the trading Member.

Further details relating to the MaC order type in ATP messages are given in Section 3.4 and the cancel reason codes are provided in Section 4.2.

2.5 Cancel on Disconnect

Trading Members using ATP should note that 'cancel on disconnect' behaviour is always in place on ATP sessions.

In other words, any open orders are cancelled as soon as a session disconnect is detected or if the user requests a session logout.

Note that after MaC lock time, matched order quantity is not cancelled by a disconnect or session logout.

2.6 Self-Trade Prevention

When a Member has multiple trading connections, Aquis offers the option for self-trade prevention to be configured across these trading sessions. This prevents a Member from trading with themselves by cancelling the resting order that would otherwise match.

- **Cancel Resting Order** – If an incoming (or price modified) order would trade with another order from the Member that is already on the Aquis order book, the older resting order is force cancelled. The incoming order may trade with other orders, or is posted to the order book, as normal.

Trading Members who wish to use self-trade prevention should contact Aquis support.

2.7 Post-Only Order Type

Aquis offers a post-only order type which a Member may use to ensure that an order can only be passive. If, in the process of receiving and posting the order, there are orders which it could match against then the Aquis trading system will cancel the post-only order.

A post-only order is specified using *orderType* = 9 in an Order Add message. If the order is cancelled back to prevent an aggressive trade the Order Add Response message will carry the *status* of Cancelled with *cancel reason* code 9 (post-only).

A post-only order can be cancelled like any other order.

If a post-only order is modified (for example to change price) it will only be updated on the order book if it will not trade on entry, otherwise the order will be cancelled back to the trading Member.

For further details on message formats see Section 3.4 and Section 4.2.

3 ATP Message Formats

This section provides details of the message formats used within the Aquis Trading Protocol. This includes data types, message headers, message fields and descriptions.

3.1 Data Types

In all messages, 1-byte packing is used and all integers are represented in little-endian format.

Data type	Size	Value
char(<i>n</i>)	<i>n</i>	Left justified ascii string, padded with zero (0x00) to length <i>n</i>
u8	1	unsigned integer 0 – 255
u16	2	unsigned integer 0 – 65,535
u32	4	unsigned integer 0 – 4,294,967,295
u64	8	unsigned integer 0 – $2^{64} - 1$
Price	8	unsigned integer representing price with 5 decimal places implied i.e. value 1462500 represents a price of 14.625
Time	8	unsigned integer representing elapsed time in nanoseconds since Unix epoch 00:00 UTC on 1st January 1970

3.2 Message Structure

All ATP messages carry a standard message header followed (for most message types) by a message body.

The header identifies the length and type of the message and, where appropriate, the sequence number of the message. The message body for a particular type of message is always a fixed length with all fields in a fixed order to support efficient creation and parsing of the binary ATP messages.

3.2.1 Message Header

The structure of the ATP message header is as follows:

Field name	Type	Offset	Width	Comments
Length	u16	0	2	Length of message including this header
msgType	u8	2	1	Message type
msgSeqNo	u32	3	4	Message sequence number, from the user or Aquis

3.3 Session Level Messages

All ATP session level messages are unsequenced. They carry the next sequence number that will be sent on the next business message in that stream (trading Member to Aquis or Aquis to trading Member). The value does not increment until a business level message is sent.

3.3.1 Login Message

The Login message is sent by the trading Member as the initial message used to establish a trading session. It is also used to re-establish a session after a break.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 1</i>
protocolVersion	u16	7	2	Version of the protocol being used
senderId	char(16)	9	16	Value as agreed with Aquis to identify the Member's ATP session
password	char(16)	25	16	Password as agreed with Aquis to verify the user
inactivityTimeout	u16	41	2	Optional inactivity timeout, in seconds. If there are no messages received for this period of time then Aquis will close the session (triggering cancel on disconnect)
atpSeqNo	u32	43	4	The next sequence number that the trading Member is expecting to receive on messages from Aquis

The *protocolVersion* is a two byte field to represent the version of the ATP protocol that the user conforms to, with major version number in the most significant byte and minor version number in the least significant byte.

The latest version, v2.0, is represented by value 0x0200 i.e. MSB = 2, LSB =0.

When logging in, the *atpSeqNo* is used to specify the next sequence number from Aquis that the trading Member is expecting. After a drop or break in the session, this can be used by Aquis to identify a gap and trigger any missed messages to be re-sent (as detailed in the Login Response section below).

3.3.2 Login Response Message

The Login Response message is sent by Aquis to acknowledge a Login request, and either accept or reject it.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 2</i>
resultCode	u8	7	1	Set to 0 for successful login Rejection codes are: 1 = Already Logged In 2 = Sequence Number Error 3 = Unsupported Protocol 4 = Failed Authentication (incorrect password)

clientSeqNo	u32	8	4	Next sequence number Aquis expects to receive
-------------	-----	---	---	---

If the Login request message is invalid, Aquis will set an appropriate *resultCode* in its Login Response. Note that if the *senderId* field is not recognised as a valid id for a session assigned to a known trading Member, or the Login originates from an unexpected source network, then Aquis will drop the connection without sending any response.

If the Login request is re-establishing the connection after a break, there may be messages that the trading Member has missed (cancellation of any open orders or potentially any trade reports that were in process at the time of the disconnect).

Aquis will compare its current business message sequence number with the *atpSeqNo* provided on the Login request to detect any gap, and will immediately send any missed messages to the trading Member. Aquis will then send the Login Response carrying the current (next expected) Aquis sequence number in its header. On receipt of this Login Response message, the trading Member knows that they have caught up with any missed messages and is now free to resume trading.

3.3.3 Heartbeat

A Heartbeat message is simply a message header with *msgType* is set to 0 and *msgSeqNo* set to the sequence number that will be sent on the next business message.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType</i> = 0

For example, pre-market before any orders have been sent each Heartbeat message will carry *msgSeqNo* = 1. The value does not increment because the heartbeat is an unsequenced, session level message.

Aquis will respond to a Heartbeat message with an outbound Heartbeat message to confirm receipt and the reliability of the connection.

3.3.4 Logout Request Message

The Logout Request message is used by the trading Member to request the closure of a trading session.

There is no message body required, simply a message header with *msgType* = 3.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType</i> = 3

3.3.5 Logout Message

The Logout message is usually sent as a response to the Logout Request. However it may also be sent by Aquis in case of a low sequence number or other protocol violation, or for other reasons.

The TCP/IP connection is closed immediately after this message has been sent.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 4</i>
reasonCode	u8	7	1	Reason code for logout 0 = User Requested 1 = Admin (Market Operations) 2 = Disconnect 3 = End of Day 4 = Inactivity Timeout 5 = Protocol Error 6 = Sequence Number Error
reasonText	char(32)	8	32	Text describing reason for logout

3.4 Business Messages

The ATP business messages allow for order entry, order management and trade handling.

All ATP business messages are sequenced and so the message stream can be recovered after a drop. For example, if a trading Member has a failure of their network connection then when the session is re-established they can receive confirmation that any open orders were cancelled and details of any trades that may have occurred just as the connection failed.

3.4.1 Order Add Message

The Add Order message is sent by the trading Member to enter an order for a particular security.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 5</i>
securityId	u16	7	2	Numeric security identifier
orderType	u8	9	1	1 = Limit Order 6 = Market at Close 9 = Post-Only
timeInForce	u8	10	1	Time in force for this order 1 = Day order 2 = Fill or Kill (FOK) 3 = Immediate or Cancel (IOC)
side	u8	11	1	1 = Buy Order, 2 = Sell Order
quantity	u32	12	4	Number of shares
price	u64	16	8	Limit price of the order (zero for MaC order)
orderCapacity	u8	24	1	1 = AOTC (A), 2 = DEAL (P), 3 = MTCH (R)

account	u8	25	1	Clearing account identifier 1 = House Account 2 or above= ID for an agreed Client Account code	
userTag	u64	26	8	Free form tag assigned by trading Member	
Flags	u8	49	1	Bit0 – LiqProv;	Whether the order relates to liquidity provision activity 0 = No 1 = Yes
				Bit1 – AlgoTradeFlag	Whether the order was generated by an algorithm 0 = No 1 = Yes
				Bit2 – DEAFFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
tableSelect1	u8	34	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode1	u32	35	4	Short Code (as qualified by previous field)	
tableSelect2	u8	39	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode2	u32	40	4	Short Code (as qualified by previous field)	
tableSelect3	u8	44	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode3	u32	45	4	Short Code (as qualified by previous field)	

3.4.2 Order Cancel Message

The Order Cancel message is sent when a user wishes to cancel an open order.

Field name	Type	Offset	Width	Comments	
Header		0	7	msgType = 7	
orderRef	u32	7	4	Order reference number	
userTag	u64	11	8	Free form tag assigned by trading Member	
Flags	u8	34	1	Bit0 – LiqProv;	Whether the order relates to liquidity provision activity 0 = No 1 = Yes
				Bit1 – AlgoTradeFlag	Whether the order was generated by an algorithm 0 = No 1 = Yes

				Bit2 – DEAFflag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
tableSelect1	u8	19	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode1	u32	20	4	Short Code (as qualified by previous field)	
tableSelect2	u8	24	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode2	u32	25	4	Short Code (as qualified by previous field)	
tableSelect3	u8	29	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	
shortCode3	u32	30	4	Short Code (as qualified by previous field)	

3.4.3 Order Modify Message

The Order Modify message is sent when a user wishes to modify an open order. Order quantity and/or limit price may be modified. Both values must be included, even if one of them is unchanged.

Field name	Type	Offset	Width	Comments	
<i>Header</i>		0	7	<i>msgType = 9</i>	
orderRef	u32	7	4	Order reference number	
price	u64	11	8	The new price of the order	
quantity	u32	19	4	The new order quantity	
userTag	u64	23	8	Free form tag assigned by trading Member	
Flags	u8	46	1	Bit0 – LiqProv;	Whether the order relates to liquidity provision activity 0 = No 1 = Yes
				Bit1 – AlgoTradeFlag	Whether the order was generated by an algorithm 0 = No 1 = Yes
				Bit2 – DEAFflag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
tableSelect1	u8	31	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode1	u32	32	4	Short Code (as qualified by previous field)	

tableSelect2	u8	36	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	Bit 3-0= PartyRole 0 – None 1 – ClientID 2 – Executing Decision ID 3 – Investing Decision ID
shortCode2	u32	37	4	Short Code (as qualified by previous field)	
tableSelect3	u8	41	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo 2 – Firm 3 – Person	
shortCode3	u32	42	4	Short Code (as qualified by previous field)	

3.4.4 Order Add Response Message

Aquis sends an Add Order Response message to acknowledge the receipt of an Order Add message.

The message is used to notify the trading Member whether or not their order was accepted and, if so, whether it executed (partially or fully) and whether any residual quantity has been added to the book or has been cancelled.

Field name	Type	Offset	Width	Comments
<i>Header</i>		0	7	<i>msgType</i> = 6
orderRef	u32	7	4	Order reference number
marketDataId	u32	11	4	The id of this order as seen in the Aquis market data. This value will be zero if the order is cancelled (IOC) or fully traded on entry
status	u8	15	1	Status of order and reject or cancel code (if needed). 3 most significant bits order status; lower 5 bits reject code if rejected, cancel code if unexpectedly cancelled, otherwise zero. See Section 4 for values.
tradedQuantity	u32	16	4	Number of shares traded, if any
timestamp	time	20	8	Time that the order was accepted or rejected by the Aquis system
userTag	u64	28	8	Free form tag as assigned by trading Member on the Order Add message

If the order traded on entry, i.e. *tradedQuantity* is not zero, then this message will be immediately followed by the related Trade message(s).

Note that this is correct behaviour even if the *status* is Cancelled, for example for an IOC order that partially trades. The Cancelled status confirms that the residual quantity has been cancelled back to the trading Member and the subsequent Trade message(s) provide details for the traded quantity.

3.4.5 Order Cancel Response Message

Aquis sends an Order Cancel Response to accept or reject an Order Cancel message. The message is also used if an order is cancelled by Market Operations or due to cancel on disconnect.

For Market at Close orders, the message is used to cancel unmatched order quantity at lock time or if the match is cancelled for a particular security.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 8</i>
orderRef	u32	7	4	Order reference number
requestRef	u32	11	4	Sequence number (<i>msgSeqNo</i>) of the trading Member's Order Cancel message. Note that this field will be zero in the case of a forced cancel generated by Aquis or cancel messages from the MaC.
status	u8	15	1	Status of order and cancel reason or reject code; lower 5 bits cancel reason or reject code as appropriate, 3 most significant bits order status. See Section 4 for values.
timestamp	time	16	8	Time that the cancellation was accepted or rejected by the Aquis system
userTag	u64	24	8	Free form tag as assigned by trading Member on the Order Cancel message

Note that cancel reason is particularly important for MaC orders, as discussed in Section 2.4.

3.4.6 Order Modify Response Message

Aquis sends an Order Modify Response to accept or reject an Order Modify message.

Field name	Type	Offset	Width	Comments
Header		0	7	<i>msgType = 10</i>
orderRef	u32	7	4	Order reference number
requestRef	u32	11	4	Sequence number (<i>msgSeqNo</i>) of the trading Member's Order Modify message.
status	u8	15	1	Status of order and reject code (if needed); lower 5 bits reject code (or zero if not rejected), 3 most significant bits order status. See Section 4 for values.
timestamp	time	16	8	Time that the modification was accepted or rejected by the Aquis system
userTag	u64	24	8	Free form tag as assigned by trading Member on the Order Modify message

3.4.7 Trade Message

The Trade message is published by Aquis when an order executes. The Trade message is also used to communicate indicative trade reports at lock time for Market at Close orders.

Field name	Type	Offset	Width	Comments
<i>Header</i>		0	7	<i>msgType = 11</i>
orderRef	u32	7	4	Order reference number
quantity	u32	11	4	Number of shares traded
price	u64	15	8	The price of the trade
side	u8	23	1	1 = Buy, 2 = Sell
tradeRef	u32	24	4	Trade reference number
ccpCode	u8	28	1	Clearing CCP code 1 = Self clearing 2 and above = CCP code, mapping is available from Aquis connectivity team
liqIndicator	u8	29	1	Liquidity indicator: 1 = Added liquidity 2 = Removed liquidity 3 = Removed liquidity (hidden quantity) 4 = Auction (MaC)
securityId	u16	30	2	Numeric security identifier
timestamp	time	32	8	Time that the trade occurred on the Aquis system
userTag	u64	40	8	Free form tag as assigned by trading Member on the Order Add message or the most recent Order Modify message

Note that the *price* and *tradeRef* values will be zero for indicative trade reports published when MaC orders are matched at lock time.

3.4.8 Trade Bust Message

If a trade is invalidated, a Trade Bust message is sent to notify the trading Member.

Field name	Type	Offset	Width	Comments
<i>Header</i>		0	7	<i>msgType = 12</i>
orderRef	u32	7	4	Order reference number
quantity	u32	11	4	Number of shares on invalidated trade
price	u64	15	8	The price of the invalidated trade
side	u8	23	1	1 = Buy, 2 = Sell
tradeRef	u32	24	4	Trade reference number for the invalidated trade
timestamp	time	28	8	Time that the trade was busted by Aquis

4 Status and Reason Codes

4.1 Order Status

This Order Status value is provided in the three most significant bits of the *status* byte field in the Order Add Response, Order Cancel Response and Order Modify Response messages.

Code	Order Status
1	Pending New (Internal use only)
2	Acknowledged
3	Cancelled
4	Rejected
5	Filled
6	Modified

For example 'Acknowledged' for a day order is encoded as **01000000** (0x40), whereas 'Modified' is encoded as **11000000** (0xC0).

4.2 Modification Codes

The modification codes are provided in the lower five bits of the *status* byte field in the Order Modify Response message.

Please note that these codes only relate to version ATP 1.4 and above.

Code	Cancel Reason
1	Modification accepted
2	Order cancelled as a result of modification which updates the remaining quantity to zero
3	Order cancelled as post-only order became aggressive

4.3 Cancel Reason

The Cancel Reason is provided in the lower five bits of the *status* byte field in the Order Cancel Response message. It is also provided in the Order Add Response message if the incoming order is unexpectedly cancelled on entry.

Code	Cancel Reason
1	Member request
2	Aquis forced cancel
3	Market close
4	MaC lock (cancel of unmatched quantity)
5	<i>Reserved</i>
6	MaC cancelled / aborted
7	Self-trade prevention
8	Cancel on disconnect
9	Post-only (cancel to prevent aggressive trade) – <u>ATP version 1.3 only</u> (removed from ATP 1.4)
10	Cancel residual MaC order quantity (cancelling unmatched portion of MaC order)

This is combined with the Cancelled order status, for example a cancel in response to a Trading Member's request is encoded as **01100001** (0x61) and a cancel of unmatched quantity at MaC lock time is encoded as **01100100** (0x64).

4.4 Reject Reasons

If an Order Add, or an Order Cancel or Order Modify request, is rejected then the Rejected order status is combined with a Reject Reason in the lower five bits of the *status* byte field.

The current set of reject reason codes are given below.

For example a reject because the given price does not conform to the tick table for the security would be encoded as 100**00101** (0x85).

Code	Reject Reason
1	Not Authorised To Trade
2	Invalid Quantity
3	Invalid Price
4	Unknown Security
5	Price Does Not Conform To Tick
6	Invalid Order Type
7	Invalid Side
8	Invalid Order Capacity
9	Market Is Closed
10	Halted
11	Suspended
12	Invalid TimeInForce
13	Order Not Found / Not Open
14	No Clearing In Place
15	Failed Price Range Check
16	Invalid Clearing Account
17	Not Supported (e.g. stock not enabled for MaC)
18	Max Value Exceeded
19	MaC Ended (Cancelled or Locked)
20	Drop Feed Down
23	Stock Restricted
24	Minimum Consideration
27	Invalid
29	Bad Date
30	Duplicate
31	Reject Internal