aquis

Aquis Trading Protocol (ATP) Specification

June 2023
Version 2.10.4

Contents

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	14 Dec 2017	MiFID II fields included and updated (May 2017); updated to include POCR (Sept 2018); 20 Nov 2017 first trading day in Production; description clarification for TVTIC (Dec 2017)
2.1	-	Document version 2.1 reserved for the SDP
2.2	05 Apr 2018	Addition of LIS Cross & Benchmark Cross
2.3	25 Apr 2018	Addition of Auction on Demand

2.3.1	08 Apr 2019	Addition of Benchmark Cross – TWAP;
		Addition of Aquis Europe MICs;
		Removal of Reject Reason 14 – No Clearing In Place (Nov 2018)

2.4	13 May 2019	Addition of field orderCapacity in Order Modify Message for MaC orders Update of latest ATP version in 'Login Message'
2.5	10 Jan 2020	Update from 'Aquis SAS' to 'Aquis Europe' Addition of Iceberg order type (for ATP version 2.6 and above)
2.5.1	09 Jun 2020	Clarification of self-clearing behaviour and Trade Message MaC behaviour
2.6	07 Jun 2021	MaC update to include Aquis Europe Update to Auction on Demand to include Pegged orders and Minimum Quantity (for ATP version 2.8 and above) Addition of minQty to Order Add Extended message Update to orderType and timeInForce in Order Add and Order Add Extended message 18 Aug 2021 Additional update - Clarification of displayQuantity value in Order Add Extended message
2.7	-	*Version reserved*
2.8	29 Oct 2021	Aligning of document and ATP versions Update to order Reject Reason 19 from 'MaC Ended' to 'Auction Ended'
2.8.1	11 April 2022	Update to remove references to AoD1 functionality. All AoD functionality refers to AoD2.
2.9	-	*Version reserved*
2.10.3	26 April 2022	Addition of non-display order book functionality (for ATP version 2.10 and above): Updates to Self-Trade Prevention , Order Add, Order Add Extended, Order Add Response, Order Modify Extended, Order Modify Response Removal of behaviour descriptions. Please refer to the relevant platform Rulebook for these details.
2.10.1	May 2022	Update to Self-Trade Prevention New reject code returned if there is an issue with the reference price feed
2.10.2	June 2022	Updates to the 'Flags' field on Order Add Extended, Order Add Response, Order Modify Response. Addition of 'Flags' field on the Trade Message to indicate waiver type of trade.

2.10.4	March 2023	Correction of ATP version to 2.10 Addition of Dark to Lit Sweep functionality: Updates to Order Add Extended. Update to allow modification of Price, Quantity and MinQty for AoD orders. Updates to Trade Message, Order Add Response Message and Order Modify Response Message (Bit 7). Updated design of Specification in line with Aquis rebrand.

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.

The latest Exchange version of ATP is v2.10.

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.

Members may request 'self-trade clearing suppression' from Aquis Support.
With this configuration option, if a Member trades with themselves 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. This information is carried in the *ccpCode* field. For self-clearing to work, the *orderCapacity* must be the same on both orders and the configured trading BIC for the sessions on either side should match.

2.4 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.

2.5 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 that Member that is already on the Aquis order book, the existing resting order is automatically 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.
When *OrderType = 12,* Members can also utilise *stpEnable* on Order Add Extended messages. If:

- *stpEnable* = 0: The Order will ignore the configured self-trade prevention settings. When set, the Member's Order will not be prevented from matching with their other Orders.
- *stpEnable* = 1: The Order will apply the Member's configured self-trade prevention settings. If the Order would trade with another resting order from that Member, the Member's resting

order would not be cancelled. The resting Order would be discounted from the matching process.

2.6 Post-Only and Post-Only Cancel Replace Order Types

Please note that Members must have self-trade prevention enabled in order for the POCR order type to function correctly.

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 PO/POCR order can be cancelled like any other order.

If a PO/POCR 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.

2.7 Market at Close

Orders submitted during continuous trading phase will be accepted but will not be published until the MaC opens.

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.

Members can modify the *orderCapacity* up to the point when the trade is published.
Members can use this functionality on ATP v2.4 and higher.

Further details relating to the MaC order type are given in Section 3.4 and the cancel reason codes are provided in Section 4.3.

2.8 Order Sweeps

Aquis supports three types of Order Sweeps:

- DLO – Dark (IOC) to Lit (IOC)
- DLOD – Dark (IOC) to Lit (Day)
- DLPO – Dark (IOC) to Lit (Post-Only)

The supported Order Sweeps first sweep the dark order book (AMP), and any residual quantity (full or partial) is sent to the lit order book for execution. The *timeInForce* provided will apply to the lit portion of the order only. The dark portion will be Immediate or Cancel (IOC) only.

To send an Order Sweep the member should send an Order Add Extended Message (*msgType=21)* with the *orderType* set to 12 and the '*Route2Lit*' flag set to true. For DAY and IOC sweeps the member should set the appropriate *timeInForce* value. Members can submit a Post Only Sweep order by setting both *Bit 2* (route2lit) and *Bit 3* (postOnly) to true.

Sweep Order Type	*OrderType*	*timeInForce*	*Flags: Bit 2 'route2lit'*	*Flags: Bit 3 ' 'postOnly'*
DLO	12	3	1	0
DLOD	12	1	1	0
DLPO	12	1	1	1

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(*n*)	*n*	Left justified ascii string, padded with zero (0x00) to length *n*
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

*The Aquis system is accurate to the nearest µs. This time is multiplied by 1000 to convert µs to ns.

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*	*msgType = 1*
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, v2.10, is represented by value 0x020a i.e. MSB = 2, LSB = 10.
When the Member logs in for the first time in the day, the ATP port stores the protocol version stated in the login message and will reject any subsequent login messages that state a different protocol version. This is to ensure that all ATP traffic for that day uses the same protocol version.

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*	*msgType = 2*
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) 5 = Unknown Source Network
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*	*msgType = 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*	*msgType = 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*	*msgType = 4*
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 a the connection failed.

3.4.1 Order Add Message

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

Field name	Type	Offset	Width	Comments	
Header		*0*	*7*	*msgType = 5*	
securityID	u16	7	2	Numeric security identifier	
orderType	u8	9	1	1 = Limit Order 2 = Auction on Demand (Limit Order) 3 = Auction on Demand (Pegged/Limit) 6 = Market at Close 8 = Post-Only Cancel Replace (POCR) 9 = Post-Only (PO)	
timeInForce	u8	10	1	1 = Day order 2 = Fill or Kill (FOK) 3 = Immediate or Cancel (IOC) 9 = Good For Auction (GFA) Only Day orders are valid for the Market at Close (MaC). GFA orders are only valid for the Auction on Demand (AoD).	
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 0 for MaC orders.	
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	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 – DEAFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
				Bit3 - Reserved	*Reserved*
tableSelect1	u8	35	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	36	4	Short Code (as qualified by previous field)	
tableSelect2	u8	40	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	41	4	Short Code (as qualified by previous field)	
tableSelect3	u8	45	1	Bit 7-4= PartyRoleQualifier 0 – None 1 – Algo	Bit 3-0= PartyRole 0 – None 1 – ClientID

				2 – Firm 3 – Person	2 – Executing Decision ID 3 – Investing Decision ID
shortCode3	u32	46	4	Short Code (as qualified by previous field)	

Short Code - 4-byte short code representing decision maker (mapping to long code):

0 = NONE (no client ID)
1 = AGGR (aggregated)
2 = PNAL (pending allocation)
3 = CLIENT (order instructed by client)
4 to 4,294,967,295 = Member-specified short code

3.4.2 Order Add Extended Message

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

Field name	Type	Offset	Width	Comments	
Header		*0*	*7*	*msgType = 21*	
securityID	u16	7	2	Numeric security identifier	
orderType	u8	9	1	1 = Limit Order 2 = Auction on Demand Limit Order 3 = Auction on Demand Pegged/Limit Order 12 = Non-display Aquis Matching Pool (AMP) order	
timeInForce	u8	10	1	1 = Day order 3 = Immediate or Cancel (IOC) 9 = Good For Auction (GFA) GFA orders are only valid for the AoD. Only Day and IOC orders are valid for order sweeps.	
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 Optional when *OrderType*=12, a value of 0 is interpreted as no limit price. Required for order sweeps.	
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	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 – DEAFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
				Bit3 - Reserved	*Reserved*

tableSelect1	u8	35	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	36	4	Short Code (as qualified by previous field)	
tableSelect2	u8	40	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	41	4	Short Code (as qualified by previous field)	
tableSelect3	u8	45	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	46	4	Short Code (as qualified by previous field)	
displayQuantity	u32	50	4	Display quantity for an iceberg order. Only valid for iceberg orders. For non-iceberg orders, value must be set to 0.	
minQty	u32	54	4	Minimum quantity (shares) of an order to be executed. Only valid when *OrderType* = 2, 3 or 12. A value of 0 is interpreted as no minimum quantity. If not applicable, value must be set to 0.	
Flags	u8	58	1	Bit 0 – stpEnable	0 – No restriction 1 – Self-trade prevention enabled Only valid when *OrderType*=12.
				Bit 1 – lisOnly	0 – No restriction 1 – LIS executions only Only valid when *OrderType*=12.
				Bit 2 - Routetolit	0 – Order is not routed 1 – Route to Lit
				Bit 3 – Route to lit and Post only	0 - Order is not routed 1 – Route to Lit and Post Only
				Bit 4 – 6	*Reserved*

3.4.2.1 Iceberg Orders

Members can send an iceberg order by submitting an Order Add Extended message. The *displayQuantity* field specifies the quantity of the order to be displayed, the remainder of the order quantity is held in reserve and is not visible. The display quantity of all orders at a given price level will be traded first and then the hidden quantity. When the display quantity has been fully traded it is refreshed from the reserve.

3.4.3 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	19	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 – DEAFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
				Bit3 - Reserved	*Reserved*
tableSelect1*	u8	20	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	21	4	Short Code (as qualified by previous field)	
tableSelect2*	u8	25	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	26	4	Short Code (as qualified by previous field)	
tableSelect3*	u8	30	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	31	4	Short Code (as qualified by previous field)	

* = optional for Cancels, defaults to original order value

3.4.4 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.

For MaC orders, the *orderCapacity* field can be modified up to the point when the trade is published. If the *orderCapacity* is modified, Members are expected to update their MiFID fields (tableSelect1, shortCode1,…,tableSelect3, shortCode3).

Field name	Type	Offset	Width	Comments	
Header		*0*	*7*	*msgType = 9*	
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	31	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 – DEAFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
				Bit3 - Reserved	*Reserved*
tableSelect1*	u8	32	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	33	4	Short Code (as qualified by previous field)	
tableSelect2*	u8	37	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	38	4	Short Code (as qualified by previous field)	
tableSelect3*	u8	42	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	43	4	Short Code (as qualified by previous field)	
orderCapacity*	u8	47	1	1 = AOTC (A), 2 = DEAL (P), 3 = MTCH (R)	

* = optional for Modifies, defaults to original order value

3.4.5 Order Modify Extended Message

The Order Modify Extended message is sent when a user wants to modify an open Order Add Extended. Order quantity and/or limit price and/or display quantity and/or minimum quantity may be modified. All values must be included, even if some are unchanged.

Field name	Type	Offset	Width	Comments
Header		*0*	*7*	*msgType = 22*
orderRef	u32	7	4	Order reference number
price	u64	11	8	The new price of the order. When *OrderType*=12, a value of 0 is interpreted as no limit price.
quantity	u32	19	4	The new total order quantity (number of shares)
userTag	u64	23	8	Free form tag assigned by trading Member

Flags*	u8	31	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 – DEAFlag	Whether the order originates from a Direct Electronic Access Client 0 = No 1 = Yes
				Bit3 - Reserved	*Reserved*
tableSelect1*	u8	32	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	33	4	Short Code (as qualified by previous field)	
tableSelect2*	u8	37	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	38	4	Short Code (as qualified by previous field)	
tableSelect3*	u8	42	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	43	4	Short Code (as qualified by previous field)	
orderCapacity*	u8	47	1	1 = AOTC (A), 2 = DEAL (P), 3 = MTCH (R)	
displayQuantity	u32	48	4	New display quantity of the iceberg order	
minQty	u32	52	4	New minimum quantity (shares) of an order to be executed.	

* = optional for Modifies, defaults to original order value

3.4.6 Order Add Response Message

Aquis sends an Order Add Response message to acknowledge the receipt of an Order Add or an Order Add Extended 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
Header		*0*	*7*	*msgType = 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) First 3 most significant bits - order status; Lower 5 bits - reason code (if rejected or 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.	
Flags	u8	36	1	Bit 0 – 5	*Reserved*
				Bit 6 - lis	LIS indicator set by the system 0 – Reference Price Waiver (RPW) 1 – Large in Scale (LIS) Only valid when *OrderType*=12.
				Bit 7 – DarkFlag	Indicates if an order is a dark order. For order Sweeps see section ‘4.4.6.1 Bit 7 – DarkFlag’ 0 – No 1 – Yes

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.

3.4.6.1 Bit 7 - DarkFlag

For dark order book (AMP) only orders, Bit 7 will always be set to true.

For Order Sweeps, if the order does not execute, or executes partially on the dark order book, Bit 7 (DarkFlag) on the Order Add Response Message will be set to false to indicate that this order has been swept to the lit book.

3.4.7 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*	*msgType = 8*
orderRef	u32	7	4	Order reference number
requestRef	u32	11	4	Sequence number (*msgSeqNo*) 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 reject or cancel code (if needed) First 3 most significant bits - order status; Lower 5 bits - reason code (if rejected or cancelled); 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.

3.4.8 Order Modify Response Message

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

Field name	Type	Offset	Width	Comments	
Header		*0*	*7*	*msgType = 10*	
orderRef	u32	7	4	Order reference number	
requestRef	u32	11	4	Sequence number (*msgSeqNo*) of the trading Member's Order Modify message.	

status	u8	15	1	Status of order and reject or cancel code (if needed)
				First 3 most significant bits - order status;
				Lower 5 bits - reason code;
				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.	
Flags	u8	32	1	Bit 0 – 5	*Reserved*
				Bit 6 - lis	LIS indicator set by the system 0 – Reference Price Waiver (RPW) 1 – Large in Scale (LIS) Only valid when *OrderType*=12.

				Bit 7 - DarkFlag	Indicates if an order modify request was made for a Dark order. 0 – No 1 – Yes

3.4.9 Iceberg Order Refresh Message

Aquis sends an Iceberg Order Refresh message to notify the Member when an iceberg order is refreshed from the non-displayed reserve quantity. This new displayed quantity is published on Aquis market data as a new order. The message displays the market data ID from the original order (*origAqxOrdId)* as well as the new market data ID for the new displayed quantity (*newAqxOrdId)*, for tracking purposes against the market data feed. Note that the *orderRef* and *origAqxOrdId* applies to the iceberg order as a whole and remains unchanged for the lifetime of the order.

Field name	Type	Offset	Width	Comments
Header		*0*	*7*	*msgType = 23*
orderRef	u32	7	4	Original order reference number
origAqxOrdId	u32	11	4	The ID of the original order as seen in the Aquis market data.
newAqxOrdId	u32	15	4	The ID of the iceberg order refresh as seen in the Aquis market data.
quantity	u32	19	4	Refreshed display quantity of the iceberg order

3.4.10 Trade Capture Message

Aquis uses a Trade Capture Report message for the purposes of allowing members to submit Large In Scale (LIS) and Benchmark Cross trades to the exchange.

Field name	Type	Offset	Width	Comments	
Header		*0*	*7*	*msgType = 17*	
quantity	u32	7	4	The trade quantity	
price	u64	11	8	The price at which the trade was crossed	
securityID	u32	19	4	Numeric security identifier	
tradeCaptureType	u8	23	1	1 – LIS Cross 2 – LIS Cross (No Clearing) 3 – VWAP Benchmark Cross 4 – VWAP Benchmark Cross (No Clearing) 5 – TWAP Benchmark Cross 6 – TWAP Benchmark Cross (No Clearing)	
Flags*	u8	24	1	*Bit0 – Reserved*	*Reserved*
				Bit1 – AlgoTradeFlag	Whether the order was generated by an algorithm 0 = No 1 = Yes
				Bit2 – Reserved	*Reserved*

				Bit3 – Reserved	*Reserved*
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 as assigned by trading Member	

* = optional AlgoTradeFlag, defaults to '0 = No'

3.4.11 Trade Capture Response Message

Aquis sends a Trade Capture Response Message to accept or reject a Trade Capture Message.

Field name	Type	Offset	Width	Comments
Header		*0*	*7*	*msgType = 18*

status	u8	7	1	Status of order and reject or cancel code (if needed)
				First 3 most significant bits - order status;
				Lower 5 bits - reason code;
				See Section 4 for values.

tradeRef	u32	8	4	Trade reference number. Corresponds to the TVTIC (RTS 24 Art.12 & RTS 22 Field 3 Table 2).
requestRef	u32	12	4	Sequence number (*msgSeqNo*) of the trading Member's Trade Capture Message.
userTag	u64	16	8	Free form tag as assigned by trading Member on the trade capture message.

3.4.12 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
Header		*0*	*7*	*msgType = 11*
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. Corresponds to the TVTIC (RTS 24 Art.12 & RTS 22 Field 3 Table 2).
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) 6 = Auction on Demand (AoD)

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.	
Flags	U8	48	1	Bit 0 – 5	*Reserved*
				Bit 6 - lis	Indicates the pre trade transparency waiver under which the trade was conducted. 0 – Reference Price Waiver (RPW) 1 – Large in Scale (LIS) Only valid when *OrderType*=12.
				Bit 7 - DarkFlag	Indicates whether a trade was executed on the dark order book. 0 – No 1 – Yes

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

3.4.13 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
Header		*0*	*7*	*msgType = 12*
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

4.3 Order Status

The Order Status code 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 **010**00000 (0x40), whereas 'Cancelled' is encoded as **011**00000 (0x60).

4.4 Modification Reasons

The Modification Reason code is 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	Modification Reason
1	Modification accepted
2	Order cancelled as a result of modification which updates the remaining quantity to zero

4.5 Cancel Reasons

The Cancel Reason code 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	Expired
5	*Reserved*
6	Aborted
7	Self-trade prevention

8	Cancel on disconnect
9	Post-only cancel (cancel to prevent aggressive trade)
10	Cancel residual quantity
11	Post-only cancel resting (cancel to prevent aggressive trade)
12	Minimum resting value

This is combined with the Cancelled order status, for example a cancel in response to a Trading Member's request is encoded as 011**00001** (0x61) and a cancel of unmatched quantity at MaC lock time is encoded as 011**00100** (0x64).

4.6 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. *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
15	Failed Price Range Check
16	Invalid Clearing Account
17	Not Supported
18	Max Value Exceeded
19	Auction Ended
20	Drop Feed Down
21	Technical Reject
23	Stock Restricted

Code	Reject Reason
24	Minimum Consideration
27	Invalid
29	Bad Date
30	Duplicate
31	Reject Internal