



ASX Trade24 Developer's Guide FIX Specifications

Version Number
Effective Date

1.5
25 May 2014

Table of Contents

1	OVERVIEW	4
1.1	SUPPORT	4
1.2	REVISION HISTORY	4
2	INTRODUCTION TO FIX	5
3	MESSAGE FORMAT AND DELIVERY	6
3.1	MESSAGE FORMAT	6
3.2	DATA TYPES	6
3.3	SEQUENCE NUMBERS	7
3.4	HEARTBEATS	7
3.5	ORDERED MESSAGE PROCESSING	7
3.6	POSSIBLE DUPLICATES	7
3.7	DATA INTEGRITY	8
3.8	REQUIRED AND OPTIONALLY-REQUIRED FIELDS	8
3.9	MESSAGE ACKNOWLEDGMENT	8
3.10	USER DEFINED FIELDS	8
4	SESSION PROTOCOL	9
4.1	FIX SESSION DEFINITION	9
4.2	LOGON	9
4.3	MESSAGE EXCHANGE	10
4.3.1	<i>API Initialisation</i>	10
4.3.2	<i>Price, Depth and At-Best Subscriptions</i>	10
4.3.3	<i>New Order and Order Modification requests</i>	10
4.4	LOGOUT	11
4.5	MESSAGE RECOVERY	11
4.5.1	<i>Message Loss</i>	11
4.5.2	<i>Gateway to Host Connection Loss and Recovery</i>	11
4.5.3	<i>API to Gateway Connection Loss and Recovery</i>	11
4.5.4	<i>Resend Process</i>	12
4.6	RESPONSES BY MESSAGE TYPE IF A SEQUENCE NUMBER MISMATCH DETECTED	12
4.7	ORDER BEHAVIOUR BY EXECINST TYPE	13
5	FIX MESSAGES	14
5.1	MESSAGE HEADER	14
5.2	MESSAGE TRAILER	14
5.3	ADMINISTRATIVE MESSAGES	15
5.3.1	<i>Heartbeat</i>	15
5.3.2	<i>Logon Message</i>	16
5.3.3	<i>Test Request</i>	17
5.3.4	<i>Resend Request</i>	18
5.3.5	<i>Reject</i>	19
5.3.6	<i>Sequence Reset - Gap Fill and MsgSeqNum Reset</i>	20
5.3.7	<i>Logout</i>	21
5.4	APPLICATION MESSAGES	22
5.4.1	<i>Base Contract (Instrument) Download Request</i>	23
5.4.2	<i>Base Contract Download Record</i>	23
5.4.3	<i>Contract Parameter Download Request</i>	24
5.4.4	<i>Contract Parameter Download Record</i>	24
5.4.5	<i>Market State Download Request</i>	25
5.4.6	<i>Filtered Market State Download Request</i>	25
5.4.7	<i>Market State Download Record</i>	26
5.4.8	<i>Trade/Order Download Request</i>	28
5.4.9	<i>Trade/Order Download Complete</i>	28
5.4.10	<i>Filtered Market Update Request</i>	29

5.4.11	<i>Market Update Record</i>	29
5.4.12	<i>Custom Market Update Record</i>	31
5.4.13	<i>At-Best Request</i>	32
5.4.14	<i>At-Best Update Record</i>	32
5.4.15	<i>Depth of Market Request</i>	33
5.4.16	<i>Depth of Market Update Record</i>	33
5.4.17	<i>User Text Message</i>	34
5.4.18	<i>Request for Quote Message</i>	34
5.4.19	<i>User Defined Request Reject</i>	35
5.4.20	<i>New Order</i>	36
5.4.21	<i>New Custom Order</i>	37
5.4.22	<i>Execution Reports</i>	39
5.4.23	<i>Custom Order Execution Report</i>	41
5.4.24	<i>Update Request</i>	43
5.4.25	<i>Order Cancel Request</i>	44
5.4.26	<i>Order Cancel Reject</i>	45
5.4.27	<i>Trade Update Request</i>	46
6	FIELD DEFINITIONS	47
7	APPENDIX A - CHECKSUM CALCULATION	59
8	FIELD INDEX SORTED BY TAG NUMBER	60
9	MESSAGES NOT IMPLEMENTED	62
10	UNUSED FIELDS SORTED BY FIELD TAGS	63

1 Overview

ASX Trade24 is the ASX's proprietary trading platform that supports ASX 24 markets and products/instruments. ASX Trade24 operates on a 24/6 basis, offering debt, equity index, commodities and Contract for Difference (CFD) products with a full suite of trading/order management functionalities.

The *ASX Trade24 Administrator's Guide* provides Participant administrators with information on the full suite of ASX Trade24 non-trading functionalities and access methodology.

The *ASX Trade24 Developer's Guide – FIX Specification* contains the relevant technical details which allow third party applications to connect and interact with ASX Trade24.

The *ASX Trade24 Developer's Guide – Markets & Functionalities* provides information on the functionality and behaviour of ASX Trade24.

Developers should utilise these documents in tandem during the design, validation and implementation stages of Application Program Interface (API) deployment.

1.1 Support

The ASX Market Access team operates a "service desk" style support centre for the customers of the ASX. ASX Market Access provides support coverage during business hours of 08:00 to 18:00 (AEST) with afterhours support and escalation via the ASX Trading Operations and Markets team.

Any questions in relation to ASX Trade24 should be directed to ASX Market Access:

Email: MarketAccess@asx.com.au
Tel (Dom): 1800 663 053
Tel (Int): + 61 2 9227 0372

1.2 Revision History

Date	Author	Notes
5 Jul 2012		Version 1.4
11 Feb 2014	DH	Added new values for Tag 103 and Tag 5014

2 Introduction to FIX

The Financial Information Exchange (FIX) Protocol is a message standard developed to facilitate the electronic exchange of information related to securities transactions. It is intended for use between trading partners wishing to automate communications.

The message protocol, as defined, will support a variety of business functions. FIX was originally defined for use in supporting US domestic equity trading with message traffic flowing directly between principals. As the protocol evolved, it was expanded to allow third parties to participate in the delivery of messages between trading partners. As subsequent versions of FIX are released it is expected that functionality will continue to expand.

Although this protocol was originally intended for securities trading, it has been adopted as the message protocol for the ASX Trade24 Gateways (Gateways) as it is now recognised as an industry standard by commodity exchanges. However, given that it did start out as a securities protocol, a fair amount of customisation is needed to accommodate commodities trading and also to accommodate the trading conventions and practices within ASX Trade24 itself.

Version 4.0 of the FIX specification has been used as the starting point. The ASX Trade24 implementation of FIX is comprised of a subset of the existing FIX 4.0 message types and a set of new (ASX Trade24 only) message types.

3 Message Format and Delivery

The FIX protocol is defined at two levels; session and application. The session level is concerned with the delivery of data while the application level defines business related data content. The following section summarizes general specifications for constructing and transmitting FIX messages.

3.1 Message Format

The general format of a FIX message is a standard header followed by the message body fields and terminated with a standard trailer.

Each message is constructed of a stream of <tag>=<value> fields.

Except where noted, fields within a message can be defined in any sequence (i.e. relative position of a field within a record is inconsequential); exceptions are explicitly defined otherwise (certain header fields and fields within repeating data groups).

All fields in a FIX message are terminated by a delimiter character; the non-printing, American Standard Code for Information Interchange (ASCII) "SOH" (#001), is used for this purpose. Records are delimited by the "SOH" character following the CheckSum field; all records begin with the "8=FIX.4.0" string and terminate with "10=nnn<SOH>".

There shall be no embedded delimiter characters within fields except for RawData, tag 96.

3.2 Data Types

Data types are mapped to ASCII strings as follows:

- int: Sequence of digits without commas or decimals and optional sign character (ASCII characters "-", "0" - "9"). The sign character utilizes one byte (i.e. positive int is "99999" while negative int is "-99999").

Examples:

723 in field 21 would be mapped int as |21=723|.

-723 in field 12 would be mapped int as |12=-723|

- float: Sequence of digits with optional decimal point and sign character (ASCII characters "-", "0" - "9" and "."); the absence of the decimal point within the string will be interpreted as the float representation of an integer value. All float fields must accommodate up to fifteen significant digits.
- char: Alpha-numeric free format strings can include any character or punctuation except the delimiter. All char fields are case sensitive (i.e. morstatt ≠ Morstatt).
- time: Time/date combination in YYYYMMDD-HH:MM:SS format, colons and dash required. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-59.
- date: Date in YYYYMMDD format. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
- data: Raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. Caution: may contain the delimiter (SOH) character.

3.3 Sequence Numbers

All FIX messages are identified by a unique sequence number. Sequence numbers are initialized at the start of each FIX session (see Session Protocol section) starting at 1 (one) and increment throughout the session. Monitoring sequence numbers will enable parties to identify and react to missed messages and to gracefully synchronize applications.

The API will establish an independent incoming and outgoing sequence series. The API will maintain a sequence series to assign to outgoing messages to the Gateway and a separate series to monitor for sequence gaps on incoming messages from the Gateway.

3.4 Heartbeats

During periods of message inactivity both the API and the Gateway will generate Heartbeat messages at regular time intervals (1 sec). The heartbeat is useful for monitoring the status of the communication link and to identify incoming sequence number gaps.

3.5 Ordered Message Processing

The FIX protocol assumes complete ordered delivery of messages between parties. When dealing with message gaps, all messages subsequent to the last message received will be requested. For example if the receiver misses the second of five messages sent, messages 3-5 will be ignored and a resend request generated for messages 2-5. If more messages (i.e. messages 6 & 7) were received after the resend request but before the first resent message (i.e. message 2) these messages will be ignored by the receiver and resent by the sender.

3.6 Possible Duplicates

When the API or Gateway is responding to a resend request a possible duplicate (PossDupe) message is generated. The PossDupe will be a retransmission (with the same sequence number, MsgSeqNo) of the application data in question with the PossDupFlag included and set to "Y" in the header.

It is the receiving application's responsibility to handle the PossDupe message (i.e. treat as a new message or discard as appropriate). All messages created as the result of a resend request will contain the PossDupFlag field set to "Y"; messages with the PossDupFlag field set to "N" should be treated as original transmissions.

Note: When retransmitting a message with the PossDupFlag set to Y, it is always necessary to recalculate the CheckSum value. The only fields that can change in a PossDup message are the CheckSum, OrigSendingTime, SendingTime, BodyLength and PossDupFlag.

3.7 Data Integrity

FIX allows the integrity of message data content to be verified in two ways; verification of record length and a simple checksum of characters.

The record length is indicated in the BodyLength field and is verified by counting the number of characters in the message following the BodyLength field up to, and including, the delimiter immediately preceding the CheckSum tag ("10=").

The CheckSum integrity check is calculated by summing the binary value of each character from the "8" of "8=" up to and including the <SOH> character immediately preceding the CheckSum tag field and comparing the least significant eight bits of the calculated value to the CheckSum value (see Appendix A for a complete description for calculating the CheckSum field).

3.8 Required and Optionally-Required Fields

Each message within the protocol is comprised of required and conditionally required (fields which are required based on the presence or value of other fields) fields. The API should be designed to operate when only the required fields are present and when the conditionally required fields are also present.

3.9 Message Acknowledgment

The FIX session protocol is based on an optimistic model; normal delivery of data is assumed (i.e. no acknowledgment of individual messages) with errors in delivery identified by message sequence number gaps. Each is identified by a unique sequence number. It is the receiving application's responsibility to monitor incoming sequence numbers to identify message gaps for response with resend request messages.

The FIX protocol does not support individual message acknowledgment. However, a number of application messages require explicit application level acceptance or rejection. Orders, cancel requests and cancel/replace requests require specific application level response.

3.10 User Defined Fields

In order to provide maximum flexibility for its users, the FIX protocol accommodates User Defined Fields.

User defined field tags 5001 – 5059²⁵ have been used in the implementation of ASX Trade24.

4 Session Protocol

4.1 FIX Session Definition

A FIX session is defined as a bi-directional stream of ordered messages between two parties within a continuous sequence number series. A single physical connection will support a single FIX session and is assumed that parties will connect once and remain active throughout the ASX Trade24 trading session.

Recovery logic will be implemented for line or machine failures which may require disconnect/reconnect of the API. If the API connection to the Gateway is lost or disconnected, the FIX Session is terminated. Upon reconnection, a new FIX session is established.

The FIX session protocol is based on an optimistic model; normal delivery of data is assumed (i.e. no communication level acknowledgment of individual messages) with errors in delivery identified by message sequence number gaps. This section provides details on the implementation of the FIX session layer and dealing with message sequence gaps.

The following terms are used throughout this section:

- **Valid FIX Message** is a message that is properly formed according to this specification and contains a valid body length and checksum field
- **API** is the application which establishes the telecommunications link and initiates the FIX Session via transmission of the initial Logon message.
- **Gateway** is the receiver of the FIX session. The Gateway has responsibility to perform authentication and formally declare the connection request “accepted” through transmission of an acknowledgment Logon message.

An ASX Trade24 FIX session is comprised of at least three parts:

- Logon
- Message Exchange, such as:
 - API Initialisation
 - Price, Depth and At-Best Subscriptions
 - New Order and Order Modification requests
 - Other requests
- Logout.

4.2 Logon

Establishing a FIX connection involves four distinct operations:

1. The API establishes a TCP socket to port 2634 on the Gateway
2. The API sends a Logon message
3. The Gateway authenticates the identity of the API
4. The Gateway replies with a confirmation Logon message, which confirms a FIX session has been established successfully.

Note: As soon as the FIX session has been established, the API should be ready to receive and respond to any messages received from the Gateway.

4.3 Message Exchange

After completion of the **Logon** process, the **message exchange** process begins. The formats for all valid messages are detailed in the sections 'Administrative Messages' and 'Application Messages'.

4.3.1 API Initialisation

Once an API has successfully logged onto the Gateway, it should obtain the current Instrument, Contract, Market State, Trade History, and Active Order information in order to initialise with current and valid information.

It is recommended that the API performs a full initialisation after every logon by requesting all of the downloads from the Gateway. Failure to do so may result in the API having an incomplete or incorrect picture of ASX Trade24 Instruments and their associated tradeable contracts.

The downloads are:

1. **Base Contract Download** – Snapshot of Instrument information, including Exchange code, Commodity code, Currency and if the Instrument has Futures contracts only, Options only, or both.
2. **Contract Parameter Download** – Snapshot of Contract information, including Month-Year of the contract, whether the contract is a Future or Option, the Tick Increment of pricing points, and the Fractional Indicator for calculation of the contract price.
3. **Market State Download / Filtered Market State Download** – Snapshot of Price and Instrument State information such as, Instrument status (pre-open, open, etc.), associated contract(s) top-of-market price information including Bid/Ask/Open/High/Low/Last/Close, and Total Traded Volume.
4. **Trade/Order Download** – Snapshot of current Trades and active Orders, including shared orders.

Note: If an API logs off and back on, the old FIX session is terminated and a new FIX session is established. It is incorrect to perform a Resend Request after logon to obtain any missed updates that may have occurred whilst the API was not connected to the Gateway.

Every time the API logs onto the Gateway, it should always perform a full initialisation by requesting the downloads listed above.

4.3.2 Price, Depth and At-Best Subscriptions

In order to receive real-time updates of market activity, the API must request one or more subscriptions. The ASX Trade24 subscription types are:

1. **Filtered Market Update Request** – Provides a stream of real-time updates such as top-of-market price information (Bid/Ask/Open/High/Low/Last/Close), Instrument Status information and Total Traded Volume.
2. **At Best Request** – Provides a stream of real-time updates that show the individual volumes of the orders at the Best Bid and Best Ask price. An At-Best update may contain up to 10 bid orders and 10 ask orders at the best Bid/Ask price. The At-Best configuration varies by Instrument.
3. **Depth of Market Request** – Provides a stream of real-time updates that contain up to 5 price levels, and the bid and ask volume at each price level. The Depth of Market configuration varies by Instrument.

4.3.3 New Order and Order Modification requests

Depending on the Trading Rights of the TradingID assigned to the API, the API may enter new orders, modify existing orders, or cancel orders. (See Section 5 For further information)

4.4 Logout

Normal termination of the FIX session will be completed via the exchange of Logout messages. Termination by other means should be considered an abnormal condition and dealt with as an error.

Before actually closing the session, the Logout initiator should wait for the opposite to respond with a confirming Logout message. This gives the remote end a chance to perform any Gap Fill operations that may be necessary. The session may be terminated if the remote side does not respond in a 10 second timeframe.

Note: Log out does not affect the state of orders flagged as Purge. For Details on Order Purge/Retain behaviour, refer to section 4.7 Order Behaviour by Execlnst type.

4.5 Message Recovery

The following sections provide details on how to recover messages in the event of missing message(s) when the FIX connection is maintained and recovery where the FIX connection is lost.

4.5.1 Message Loss

During initialisation, or in the midst of a FIX session, message gaps may occur which can be detected by monitoring incoming sequence numbers. The API must maintain two sequence number queues (Incoming + Outgoing) for each FIX session. With each message sent/received, the sequence number queues should increment by 1 (one). The API should use the MsgSeqNum received in the logon acknowledgement as its starting point for monitoring incoming messages from the Gateway

Both sequence number queues are usually, but not always, initialised at 1 (one) at the beginning of the FIX session. If the API does not start at 1, the API's developer(s) should be aware of the MsgSeqNum field's upper limit, specified in Section 6 - Field Definitions. Once this limit is reached, the FIX session must be terminated and a new session established.

If the incoming sequence is greater than the expected sequence number, this indicates that 1 or more messages have been missed and the API needs to request a retransmission of the relevant message(s) via a Resend Request.

Upon receipt of a Resend Request the Gateway will respond with the requested messages (in order) with their original MsgSeqNum numbers and PossDupFlag set to "Y". If the Gateway is responding to a Resend Request then the API is responsible for checking if the message is a duplicate. Similarly, if the API is responding to a Resend Request then the Gateway is responsible for checking if the message is a duplicate.

Note: The process of resending and synchronizing messages is referred to as "gap filling".

4.5.2 Gateway to Host Connection Loss and Recovery

If the Gateway loses connection to the ASX Trade24 host, the API will be disconnected from the Gateway. Any active orders that were flagged as Purge are cancelled. Any orders flagged as Retain will not be cancelled.

Upon re-establishing a FIX session, the API should perform the standard API initialisation process, and request the subscriptions it requires, as outlined in Section 4.3 - Message Exchange.

4.5.3 API to Gateway Connection Loss and Recovery

If the API loses its connection to the Gateway, or Logs Off the Gateway, any active Purge orders are Not cancelled.

Upon re-establishing a FIX session, the API should perform the standard API initialisation process, and request the subscriptions it requires, as outlined in Section 4.3 - Message Exchange.

4.5.4 Resend Process

During the resend process, certain administrative messages should not be retransmitted. Instead, a special SeqReset-GapFill message is generated. The administrative messages which are not to be resent are: Logon, Logout, ResendReq, HeartBeat, TestReq and SeqReset. The SeqReset-GapFill can also be used to skip application messages that the sender chooses not to retransmit (e.g. aged orders).

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 NewSeqNum field of the GapFill message itself contains the sequence number of the highest administrative message in this group plus 1. For example, during a Resend operation there are 7 administrative messages in a row waiting to be resent. They start with sequence number 9 and end with sequence number 15. Instead of transmitting 7 Gap Fill messages, a SeqReset-GapFill message may be sent. The sequence number of the Gap Fill message itself is set to 9 because the remote side is expecting that message next. The NewSeqNum field of the GapFill 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 in the case where the incoming sequence number is greater than the expected incoming sequence number.

Note: Resends cannot be used to recover messages from a previous FIX session. When there is no session in progress, transactions will not be sent out over fix nor assigned a sequence number. Therefore, resends are only for messages missed within a FIX session.

When the API logs on, the first sequence number sent by the Gateway is contained in the logon acknowledgement. The API should not attempt to access messages prior to the logon acknowledgement by requesting a resend. The API should perform a full initialisation as outlined in Section 4.3 - Message Exchange to build a picture of the market and its order book.

4.6 Responses by Message Type if a Sequence Number Mismatch Detected

Message Type	Action to Be Taken if MsgSeqNum mismatch detected
Logout (35=5)	<p>If a message gap was detected, issue a ResendRequest to retrieve all missing messages followed by a Logout message which serves as a confirmation of the logout request. DO NOT terminate the session. The initiator of the Logout sequence has responsibility to terminate the session. This allows the Logout initiator to respond to any ResendRequest generated.</p> <p>Once the Logout confirmation has been received, the Logout initiator should immediately terminate the FIX session.</p> <p>(The only exception to the “do not terminate the session” rule is for an invalid Logon attempt. The Gateway has the right to terminate the session immediately. This minimizes the threat of unauthorized connection attempts.)</p>

ResendRequest (35=2)	Perform the Resend processing first, followed by a ResendRequest of your own in order to fill the incoming message gap.
SeqReset-Reset (35=4 and 123=N)	Ignore the incoming sequence number. The NewSeqNum field of the SeqReset message will contain the sequence number of the next message to be transmitted.
SeqReset-GapFill (35=4 and 123=Y)	Send a ResendRequest back. Gap Fill messages behave similar to a SeqReset message. However, it is important to insure that no messages have been inadvertently skipped over. This means that GapFill messages must be received in sequence. An out of sequence GapFill is an abnormal condition
All Other Messages	Perform Gap Fill operations.

Note: A Resend Request should never occur as a result of receiving the Logon Acknowledgement message. The API should use the MsgSeqNum received in the logon acknowledgement as its starting point for monitoring incoming messages from the Gateway.

4.7 Order Behaviour by ExecInst type

The ExecInst field is used to define order behaviour if the Gateway loses connection to the ASX Trade24 Host.

If the Gateway loses connection to the Host, orders flagged as purge (18=P) will be cancelled by the Host. Orders that are flagged as retain (18=R) will not be cancelled.

If the API loses its FIX connection to the Gateway or logs out of the FIX session, and the Gateway maintains its connection to the Host, orders marked as Purge are NOT cancelled.

Note: API developers should be aware that if the Gateway also loses connection to the Host, the Gateway will disconnect the API and terminate the FIX session, and orders flagged as purge (18=P) ARE cancelled by the Host.

5 FIX Messages

5.1 Message Header

Each message, administrative or application, is preceded by a standard header. The header is used to identify the message type, length, destination, sequence number, origination point and time.

The PossDupFlag is set when resending a message as the result of a session level event (i.e. the retransmission of a message reusing a sequence number). When received the action to be taken is:

- 1) If a message with this sequence number has been previously received, then ignore the message,
- 2) else, process the message normally.

The field, SessionNo, is not used by the Gateway to process FIX messages, or handling resend logic, however the Gateway will increment the field when the Gateway restarts. The API may choose to increment this value, such as with each new FIX logon to the Gateway. Do not use this field for FIX message recovery logic. It may be used optionally to distinguish message sessions. If choosing not to use this field, it may be permanently set to the value of 1.

Tag	Field Name	Req'd	Comments
8	BeginString	Y	FIX.4.0
9	BodyLength	Y	
35	MsgType	Y	
49	SenderCompID	Y	ASX Assigned Firm Code/Acronym
34	MsgSeqNum	Y	
5006	SessionNo	Y	Session Number
50	SenderSubID	Y	ASX Assigned Trader ID
43	PossDupFlag	Y	Always required for retransmissions, whether prompted by the sending system or as the result of a resend request.
52	SendingTime	Y	Time (in UTC/GMT) message was sent from the Gateway to the API.
122	OrigSendingTime	Y	Required for message resends, indicates time (in UTC/GMT) of original message transmission. If data is not available set to same value as SendingTime

5.2 Message Trailer

Each message, administrative or application, is terminated by a standard trailer. The trailer is used to segregate messages and contains the three digit character representation of the Checksum value. Refer to Appendix A for sample code to calculate the CheckSum field.

Tag	Field Name	Req'd	Comments
10	CheckSum	Y	

5.3 Administrative Messages

The administrative class of messages is intended to address the utility needs of the protocol. The following section describes the use of each message and provides the message layout.

Administrative messages will be generated from both sides of the connection.

5.3.1 Heartbeat

- The Heartbeat is useful for monitoring the status of the communication link and to identify when the last of a string of messages was not received.
- The Gateway will send one heartbeat per second. The API should also heartbeat once per second
- When either end of a FIX connection has not sent any data for 1 second, it will transmit a Heartbeat message. When either end of the connection has not received any data for 10 seconds, it will transmit a Test Request message. If there is still no Heartbeat message received after 10 seconds then the connection should be considered lost and corrective action should be initiated.
- Heartbeats issued as the result of Test Request must contain the TestReqID transmitted in the Test Request message. The TestReqID is used to verify that the opposite application is generating the heartbeat as the result of Test Request and not a normal timeout.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 0
112	TestReqID	Y	Required when the heartbeat is the result of a Test Request message. Format: HHMMSS (should match the TestRequest that was received)
	Standard Trailer	Y	

5.3.2 Logon Message

- The logon message is utilised to authenticate an API attempting to establish a connection from a remote system. The logon message must be the first message sent by the application requesting to initiate a FIX session.
- The HeartBtInt (108) field is used to declare the timeout interval for generating heartbeats. The Gateway will specify the heartbeat interval, set to 1 second. The API will receive the heartbeat interval in the logon acknowledgment message.
- Upon receipt of a Logon message the Gateway will authenticate the API and issue a Logon message as acknowledgment that the connection request has been accepted.
- The API must be prepared to immediately begin processing messages after receipt of the Logon.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = A
108	HeartBtInt	Y	set to 1 second, only present in the logon acknowledgment message.
95	RawDataLength	Y	Length of Raw Data
96	RawData	Y	Will be Formatted As Follows TraderID=<User ID of the Gateway><SOH> Password=<Password>
	Standard Trailer	Y	

5.3.3 Test Request

- The test request message is utilised to force a heartbeat from the opposing application. The test request message is useful for checking sequence numbers or verifying communication line status. The opposite application will respond to the Test Request with a Heartbeat containing the TestReqID.
- The TestReqID is used to verify that the opposite application is generating the heartbeat as the result of Test Request and not a normal timeout. The opposite application will include the TestReqID in the resulting Heartbeat. The string to be used as the TestReqID will be a timestamp string. This string will be the time TestReqID message was sent HHMMSS.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 1
112	TestReqID	Y	The Time The TestReqID has been sent Format: HHMMSS
	Standard Trailer	Y	

5.3.4 Resend Request

- The resend request is sent by the receiving application to initiate the retransmission of messages. This function is utilised if a sequence number gap is detected, if the receiving application lost a message, or as a function of the initialization process.
- The resend request can be used to request a single message, a range of messages or all messages subsequent to a particular message.
- 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-GapFill message is used to skip individual messages that a sender does not wish to resend.)
- The receiving application should process messages in sequence order, e.g. if message number 7 is missed and 8-9 received, the application should ignore 8 and 9 and ask for a resend of 7-9.
- To request a range of messages, BeginSeqNo = first message of range, EndSeqNo = last message of range.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 2
7	BeginSeqNo	Y	
16	EndSeqNo	Y	
	Standard Trailer	Y	

5.3.5 Reject

- The reject message should be issued when a message is received which cannot be passed through to the application level. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&), which successfully passes CheckSum and BodyLength checks.
- Rejected messages are logged and the incoming sequence number incremented.
- When a message is received which is garbled, cannot be parsed or fails a data integrity check, the Gateway will disregard the message. Processing of the next valid FIX message will cause detection of a sequence gap and a Resend Request is generated.
- Generation and receipt of a Reject message should be dealt with as an indication of a serious error by both parties as it may be the result of faulty logic in either the sending or receiving application.
- If the sending application chooses to retransmit the rejected message it should be assigned a new sequence number.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 3
45	RefSeqNum	Y	MsgSeqNum of rejected message
58	Text	Y	Where possible, message to explain reason for rejection
	Standard Trailer	Y	

5.3.6 Sequence Reset - Gap Fill and MsgSeqNum Reset

- The sequence reset message (35=4 and 123=N) is used by the sending application to reset the incoming sequence number on the receiving application to be the next sequence number to be transmitted.
- 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).
 - During normal resend processing a number of administrative messages are not resent, the Sequence Reset message is used to fill the sequence gap created.
 - In the event of an application failure it may be necessary to force synchronization of sequence numbers on the sending and receiving sides
- The sending application will initiate the sequence reset. The message is used to reset the value of the next sequence number to be transmitted.
- If the GapFill field is not present (or set to N), the sequence reset message is to recover from an out-of-sequence condition, therefore, the MsgSeqNum in the header should be ignored (i.e. the receipt of a sequence reset message with an out of sequence MsgSeqNum should not generate resend requests).
- If the Gap Fill field is present (and equal to Y), the MsgSeqNum should conform to standard message sequencing rules. This has the effect of preventing a NewSeqNum from being effective. To perform a MsgSeqNum reset ensure that 123=N.
- 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.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 4
123	GapFillFlag	N	Valid Values = Y or N If set to Y purpose is to skip a single message. MsgSeqNum reset cannot occur If set to N purpose is to perform a MsgSeqNum reset
36	NewSeqNum	Y	
	Standard Trailer	Y	

Note: the GapFill msg type (35=4 and 123=Y) cannot be used to reset the MsgSeqNo. To reset the MsgSeqNo set 123=N.

5.3.7 Logout

- The logout message is used to initiate or confirm the termination of a FIX session. Disconnection without the exchange of logout messages should be treated as a serious error.
- The logout initiator should wait for the opposite side to respond with a confirming logout message, before the initiator closes the FIX session. This gives the remote end a chance to perform any Gap Fill operations that may be necessary. The FIX session may be terminated by the logout initiator if the opposite side does not respond within 10 seconds.
- The logout initiator should not send any messages after the logout.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 5
	Standard Trailer	Y	

5.4 Application Messages

The exchange of business related information is performed via application messages. The application message is composed of the Standard Header followed by the Message Body and then the Trailer.

As the FIX.4.0 protocol does not sufficiently support the transmission of Market data, user defined messages are implemented for data transmission.

Commodity, contract and price information is requested by the API. Market downloads provide a snapshot of the market.

API Developers who are not familiar with trading on ASX Trade 24 may find it useful to note that “instruments” are not directly tradeable. “Contracts” are the tradeable identity, and are derived from their parent instrument. For further information refer to *ASX Trade24 Developer's Guide – Markets & Functionalities*.

The download of information is a three step process, as follows:

1. Base Contract (Instrument) Download
2. Contract Parameter Download
3. Market State Download

5.4.1 Base Contract (Instrument) Download Request

Base Contract Download request message is a User Defined message with the field "UserMsgCat" set to a value of 0. This message is used to request all commodity instruments.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 0 For Commodity Download Request
	Standard Trailer	Y	

5.4.2 Base Contract Download Record

After issuing a Base Contract Download Request message to the Gateway, the API should wait for Base Contract Download records. The API will be notified when the download has completed by the LastMessage field. All Contract Base records available in the system will be downloaded.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 1 For Commodity Download Record
100	ExDestination	Y	Exchange Code for the Contract
55	Symbol	Y	Commodity Code
5004	Currency	Y	The Base Currency for listed commodity
5021	FOBIndicator	Y	Indicator whether futures has futures options or both F – Futures Only O – Options Only B – Both Futures and Options
58	Text	Y	Only Present if an Error Occurred
5003	LastMessage	Y	Only Present If Current Message is the Last Message In The Download
	Standard Trailer	Y	

5.4.3 Contract Parameter Download Request

The Contract Parameter Download Request message is a User Defined message with the field "UserMsgCat" set to a value of 2. This message is used to request all Contracts listed on ASX Trade24.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 2 For Contract Parameter Download Request
	Standard Trailer	Y	

5.4.4 Contract Parameter Download Record

The Contract Parameter Download records are sent in response to a Contract Parameter Download Request.

All commodities will be transmitted to the API. The sequencing of commodities generally follows as, all futures and then option information for commodity 1, and then commodity 2, ..., then commodity n, however the sequencing is not guaranteed, and the API should be designed to process Contract Parameter Download Records if they are not received in Commodity order.

The Commodity Parameters Currency, TickIncrement, UnderlyingFuture and Status will be sent once for futures and once for options, per commodity.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 3 For Contract Parameter Record
100	ExDestination	Y	Exchange Code
55	Symbol	Y	Commodity Code
5002	InstrmntIndicator	Y	F – Futures or O – Options
5009	MonthYear	Y	Month Year Combination - Options only
5005	TickIncrement	Y	Futures or Options Price Tick Increment.
5010	UnderlyingFut	Y	Underlying Futures Contract – Only present when Tag 5002=O
5054	FractionalIndicator	Y	Fractional Indicator For Futures or Option Series.
58	Text	Y	Only Present If An Error Occurred
5003	LastMessage	Y	Last Message In the Download, only present in the last message
	Standard Trailer	Y	

5.4.5 Market State Download Request

The Market State Download request is used to request current market conditions after the API has logged onto the Gateway, and has completed contract download.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 4 For Market Download Request
	Standard Trailer	Y	

5.4.6 Filtered Market State Download Request

This record has a similar behaviour to the Market State Download Request, but it also provides the ability to filter the information that is requested. It is used to request the current market price status according to a user-selected filter of contracts. This request returns an appropriate batch of Market State Download Records matching the selected filter. A request which cannot be processed for whatever reasons will be rejected with a user-defined request reject message. This message will indicate the reason for rejection.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 11 For Filtered Market Download Request
5036	ReferenceNo	Y	API-defined reference no
5033	FilterType	Y	0 - All, 1 – By Exchange, 2 – By Instrument, 3 – By Custom Market
100	ExDestination	N	Exchange Code. Required if FilterType = 1 or FilterType = 2
55	Symbol	N	Instrument Code. Required if FilterType = 2
5034	InterSpreadsFlag	N	Include Inter-Instrument Spreads. Required if FilterType = 2. Y – include, N – do not include
	Standard Trailer	Y	

5.4.7 Market State Download Record

Market State Download Record is sent in response to a Market State Download Request. This message contains the current market information as well as contract labels for the available spread contracts. The download will consist of a set of messages for the Futures, Options, and Intra Spreads within a commodity and then all Inter Spreads.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 5 For Market Download Record
100	ExDestination	Y	Exchange Code For The Records in RawData
55	Symbol	Y	Commodity Code
5002	InstrmntIndicator	Y	F – Futures, O – Options, C - Intra Spreads or A - Inter-spreads
5028	InterSprdRatio	Y	Volume ratio for Inter Instrument Spreads (e.g. X:Y). Only present where Tag 5002 = A.
5009	MonthYear	Y	Month Year Combination (e.g. M7 for contract CCM7). Not present if 5002 = A or C.
5013	OptionsStrikeP/C	Y	Options Strike Price and Put/Call Indicator. Only present where Tag 5002=O. Note: Inclusion of 'o' after the P/C indicator indicates a Single Session Option.
5014	MarketStatus	Y	The Status of the Instrument. Not present if 5002 = A or C.
5017	SettlementPrice	Y	The Settlement Price. Only present where Tag 5014=F or L. Not present if 5002 = A or C.
5018	Volatility	Y	Options Volatility.
5019	OpenPrice	Y	Opening Trading Price; Only present if there is market activity. Not present if 5002 = C.
5020	ClosePrice	Y	Closing Price. Not present if 5002 = A or C.
5016	HighTradePrice	Y	Highest Trading Price; Only present if there is market activity. Not present if 5002 = C.
5015	LowTradePrice	Y	Lowest Trading Price; Only present if there is market activity. Not present if 5002 = C.
5022	AskPrice	Y	Current Best Ask; Only present if there is market activity.
5023	AskVolume	Y	Current Ask Volume; Only present if there is market activity.
5024	BidPrice	Y	Current Best Bid; Only present if there is market activity.
5025	BidVolume	Y	Current Bid Volume; Only present if there is market activity.
5026	LastTradedPrice	Y	Last Traded Price; Only present if there is market activity. Not present if 5002 = C.
5011	LastTradedVol	Y	Last Traded Volume; Only present if there is market activity. Not present if 5002 = C.
5055	TotalTradedVolume	N	Total Traded Volume. Only present if there is market activity. Not present if 5002 = C.

5056	InterimSettlement	N	Interim Settlement Price. Only present where Tag 5014=i, F or L. Not present if 5002 = A or C
5031	LastBuyer	N	Last Buying Firm. Only present if there is market activity. Not present if 5002 = C. Note – field is blank for all Trade24 products.
5032	LastSeller	N	Last Selling Firm. Only present if there is market activity. Not present if 5002 = C. Note – field is blank for all Trade24 products.
5057	ImpliedPrice ²⁵	N	Indicator present ONLY if prices reported are implied 1 – 5024 (BidPrice) is implied 2 – 5022 (AskPrice) is implied 3 – 5024 (BidPrice) and 5022 (AskPrice) are implied
58	Text	Y	Only on error condition
5003	LastMessage	Y	Last Message In the Download, Only present in Last Message
	Standard Trailer	Y	

5.4.8 Trade/Order Download Request

If the API needs to refresh its internal trade/order book, it may do so by requesting a Trade/Order Download.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 7 For Trade/Order Download
	Standard Trailer	Y	

The Gateway will respond with zero or more Execution Reports (MsgType = 8) with ExecTransType (Tag 20) set to "D" (Download) and will contain the active orders and/or current trades for that TraderID (including all permissible shared orders).

5.4.9 Trade/Order Download Complete

This record is used to indicate the end of a Trade/Order download

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 8 For Trade/Order Download Complete
	Standard Trailer	Y	

5.4.10 Filtered Market Update Request

This request is used to enable/disable market price updates according to a user-selected filter. When a filter is enabled the API will receive the appropriate Market Update Records in real-time. Multiple filters can be enabled at one time. Sending a request with the enable field set to N can be used to turn off updates for a particular filter.

The Gateway will not send any market price updates unless it has received one or more Filtered Market Update Requests. A request, which cannot be processed for whatever reason, will be rejected using a user-defined request reject message. This message will indicate the reason for rejection.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 10 For Filtered Market Update Request
5036	ReferenceNo	Y	MFOS-defined reference no
5033	FilterType	Y	0 - All, 1 – By Exchange, 2 – By Instrument, 3 – By Custom Market
100	ExDestination	N	Exchange Code. Required if FilterType = 1 or FilterType = 2
55	Symbol	N	Instrument Code. Required if FilterType = 2
5034	InterSpreadsFlag	N	Include Inter-Instrument Spreads. Required if FilterType = 2. Y – include, N – do not include
5035	Enable	Y	Y – enable update filter, N – disable update filter
	Standard Trailer	Y	

5.4.11 Market Update Record

The Market Update record is used to update contract prices and status in real time. These messages are sent from the Gateway to the API when there is movement in the market or when there is a change in market status (e.g. market/commodity/contract goes from open to close)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 6 For Market Update record
100	ExDestination	Y	Exchange Code
55	Symbol	Y	Commodity Code
5002	InstrumntIndicator	Y	Present for Intra spreads (C) and Inter spreads (A)
5009	MonthYear	Y	Month Year Combination (e.g. M7 for contract CCM7). Not present if 5002 = A or C.

5013	OptionsStrikeP/C	Y	Options Strike Price and Put/Call Indicator. Note: Inclusion of 'o' after the P/C indicator indicates a Single Session Option.
5012	ChangeStatusOf	Y	Required when the MarketStatus (5014) field is present. Not present if 5002 = A or C. C - Commodity (Status change is for the entire Commodity) F - Futures (Status change is for the Futures) O - Options Contract/Series (Status change is for the Option) B - Both futures and options within a Commodity M - Market (used only when the entire system has been temporarily halted (i.e. in conjunction with 5014 = Suspended) or resumed (i.e. in conjunction with 5014 = Opened) N – Single Session Option Series (used to distinguish overnight and intra-day options from regular options at a series level)
5014	MarketStatus	Y	Only present for status change. This field contains the new status while Tag 5012 ("ChangeStatusOf"), indicates at what level the new Status applies. Not present if 5002 = A or C.
5015	LowTradePrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = C.
5016	HighTradePrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = C.
5017	SettlementPrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = A or C.
5019	OpenPrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = C.
5020	ClosePrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = A or C.
5022	AskPrice	Y	Only Present When The Value For This Field Changes
5023	AskVolume	Y	Only Present When The Value For This Field Changes
5024	BidPrice	Y	Only Present When The Value For This Field Changes
5025	BidVolume	Y	Only Present When The Value For This Field Changes
5057	ImpliedPrice ²⁵	N	Indicator present ONLY if prices reported are implied 1 – 5024 (BidPrice) is implied 2 – 5022 (AskPrice) is implied 3 – 5024 (BidPrice) and 5022 (AskPrice) are implied
5026	LastTradedPrice	Y	Only Present When The Value For This Field Changes. Not present if 5002 = C.
5011	LastTradedVolume	Y	Only Present When The Value For This Field Changes. Not present if 5002 = C.
5031	LastBuyer	N	Only Present When The Value For This Field Changes. Not present if 5002 = C. Note – field is blank for all Trade24 products.

5032	LastSeller	N	Only Present When The Value For This Field Changes. Not present if 5002 = C. Note – field is blank for all Trade24 products.
5055	TotalTradedVolume	N	Total Traded Volume. Only present if volume has changed. Not present if 5002 = C.
5018	Volatility	Y	Only Present When The Value For This Field Changes. Not present if 5002 = A or C.
5056	InterimSettlement	N	Only Present When The Value For This Field Changes. Not present if 5002 = A or C
	Standard Trailer	Y	

5.4.12 Custom Market Update Record

The Custom Market Update record is used to update or notify market information associated with the custom market. These messages are sent in response to a relevant market download request or as part of filtered market update request.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 18 – Custom Market Update
5037	CustomVolume	Y	Volume available
5038	NoLegs	Y	Number of repeating legs to follow
55	Symbol	Y	Valid custom market leg contract identifier (excludes spreads)
5039	LegPrice	Y	Custom leg price
5040	LegVolumeRatio	Y	Custom leg volume ratio
5041	LegSide	Y	Custom leg side B=buy or S=sell
	Standard Trailer	Y	

Tags 55, 5039, 5040 and 5041 will be present for each leg of the Custom Order, up to a maximum of 6 legs.

5.4.13 At-Best Request

This request is used to subscribe to “at-best” market information for the selected contract/s.

Once a valid At-Best request has been sent to the Gateway, a stream of at-best update messages will be sent to the API until either the API sends a disable request or the API disconnects from the Gateway.

A request, which cannot be processed for whatever reason will be rejected using a user-defined request reject message. This message will indicate the reason for rejection.

Note: ASX Trade24 currently allows 20 At-Best subscriptions per Gateway.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 12 For At-Best Request
55	Symbol	Y	Contract Identifier
5036	ReferenceNo	Y	API-defined unique request reference no., to be applied in the responding at-best update message “ReferenceNo” field’
5035	Enable	Y	Y – enable updates, N – disable updates
	Standard Trailer	Y	

5.4.14 At-Best Update Record

The “At-Best” update record is sent from the Gateway to the API in response to an “at-best” request. This message contains current at best information for the requested contract.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 13 For At-Best Response
55	Symbol	Y	Contract Identifier
5036	ReferenceNo	Y	API-defined reference no., from the original at-best request message.
5022	AskPrice	N	Only present if there is market activity.
5024	BidPrice	N	Only present if there is market activity
5042	NoAskLevels	Y	Number of repeating ask levels to follow
5045	NoBidLevels	Y	Number of repeating bid levels to follow
5043	AskFirm	N	Required if NoAskLevels > 0. Blank due to anonymous market (** = implied order)
5044	AskVol	N	Asking firm volume (required if NoAskLevels > 0)
5046	BidFirm	N	Required if NoBidLevels > 0. Blank due to anonymous market (** = implied order)
5047	BidVol	N	Bidding firm volume (required if NoBidLevels > 0)
	Standard Trailer	Y	

5.4.15 Depth of Market Request

This request is used to subscribe to “depth of market” (DOM) information for the selected contract.

Once a valid DOM request has been sent to the Gateway, a real-time stream of DOM update messages will be sent to the API until either the API sends a disable request or the API disconnects from the Gateway.

A request, which cannot be processed for whatever reason, will be rejected using a user-defined request reject message. This message will indicate the reason for rejection.

Note: ASX Trade24 currently allows 100 DOM subscriptions per Gateway.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 14 For DOM Request
5036	ReferenceNo	Y	API-defined unique request reference no., to be applied in the responding DOM update message “ReferenceNo” field’
55	Symbol	Y	Contract Identifier
5035	Enable	Y	Y – enable updates, N – disable updates
	Standard Trailer	Y	

5.4.16 Depth of Market Update Record

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 15 For DOM Response
55	Symbol	Y	Contract Identifier
5036	ReferenceNo	Y	API-defined reference no., from the original DOM request message.
5042	NoAskLevels	Y	Number of repeating ask levels to follow
5045	NoBidLevels	Y	Number of repeating bid levels to follow
5022	AskPrice	N	Asking Price (required if NoAskLevels > 0)
5044	AskVol	N	Asking Volume (required if NoAskLevels > 0)
5024	BidPrice	N	Bidding Price (required if NoBidLevels > 0)
5047	BidVol	N	Bidding Volume (required if NoBidLevels > 0)
	Standard Trailer	Y	

5.4.17 User Text Message

The user message record is used to send text messages to the market including individual users, ASX Trade24 operations or all users. The same message is used by the API to receive user messages.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 16 For User Message
5048	SendMsgType	Y	0 – User ID (To/From) 1 – ASX 24 Operations (To/From) 2 – All Users (To)
5050	SendMsgText	Y	Text message to send/sent (maximum 60 characters)
5049	SendMsgUserId	N	Required when SendMsgType = 0
	Standard Trailer	Y	

5.4.18 Request for Quote Message

The request for quote record is used to broadcast a quote request to the market. The same message is used by the API to receive request for quote (RFQ) messages.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 17 For Quote Message
55	Symbol	Y	Contract Identifier
44	Price	Y	Quote price (not required)
5051	QuoteVolume	N	Quote volume (not required)
5052	QuoteFirm	N	Firm requesting quote (not required)
	Standard Trailer	Y	

Note: Custom market RFQ's are sent as text messages, (Tag 5001=16) with CRFQ as the leading characters in the text to indicate it is a Custom request for quote. As the text message facility only allows a maximum of 60 characters, where the leg details for the CRFQ will exceed 60 characters, the text should be split and “...” should be used at the end of the first text message to indicate an additional text message is to follow and “...” should be used at the start of the second message to indicate this forms part of a continuing message.

5.4.19 User Defined Request Reject

This record is used to reject a user-defined request message. These messages are covered by a user-defined request message with `MsgType = U`, `UserMsgCat = 9` and an appropriate value in the `UserMsgCatReq` field corresponding to the original request.

The `ReferenceNo` field is filled with the `ReferenceNo` of the original request message. The reason for rejection is indicated in the error message field.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	<code>MsgType = U</code>
5001	<code>UserMsgCat</code>	Y	Set To 9 For User-Defined Request Reject
5053	<code>UserMsgCatReq</code>	Y	Set To a <code>UserMsgCat</code> of original User-Defined Request
5036	<code>ReferenceNo</code>	Y	API-defined reference no. from the original User-Defined request message
58	Text	Y	Error message – reason request rejected
	Standard Trailer	Y	

5.4.20 New Order

- The new order message type is used by API to electronically submit orders for execution.
- New Order messages received with the PossDupFlag set in the header should be validated by ClOrdID and order parameters (side, symbol, quantity, etc.) to determine if the order had been previously submitted.
- PossDupFlags previously received should be acknowledged back to the API but not resubmitted to the host. PossDupFlags not previously received should be processed as a new order and acknowledged via an Execution - New message.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = D
11	ClOrdID	Y	API transaction number for new Orders that are submitted to Gateway. The API will use this field to locate all messages that are waiting for acceptance/rejection from the originating Gateway. This API should increment this field with every request sent to the Gateway
1	Account	Y	Account
100	ExDestination	Y	Exchange ID.
55	Symbol	Y	Contract Identifier
54	Side	Y	Bid/Ask Indicator
38	OrderQty	Y	Order Volume
40	OrdType	Y	Order Type (i.e. Limit, Fill or Kill, GTC, Timed or Market Limit)
44	Price	Y	Order Price
81	ProcessCode	Y	Can be either T(Tagged) or N(No Tag) T - Order is to be tagged when submitted to the host N - Order is not tagged when submitted to the host
18	ExecInst	Y	Can be Either R(Retained) or P(Purged) For Details, refer to section "4.7 Order Behaviour by ExecInst type"
5030	Shared	Y	Can be either S(Shared) or N(Not Shared) S - Order is shared (i.e. can be viewed, modified, cancelled) by all traders/operators in the Firm designated with Shared rights for the Share Group specified in Tag 5029 N – Order is not shared (i.e. can be viewed, modified, cancelled) only by the owner of the order.
5029	Shared Group Id.	N	Required if 5030=S. Shared group id.
58	Text	Y	Six Character Comment Field
126	ExpireTime	Y	Required if OrderType = 2
	Standard Trailer	Y	

5.4.21 New Custom Order

The New Custom Order message type is used by the API to electronically submit orders for execution in the custom market.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 19 – New Custom Market Order
11	ClOrdID	Y	API transaction number for new custom market orders that are submitted to the Gateway. The API will use this field to identify order response messages from the Gateway. The API should increment this field with every request sent to the Gateway
1	Account	Y	Account
100	ExDestination	Y	Exchange ID. For customs with legs across multiple exchanges, the order must have at least one leg with a symbol from this exchange to be valid.
38	OrderQty	Y	Order Volume
40	OrdType	Y	Order Type (i.e. Limit, Fill or Kill) Note – GTC and MLM order types not allowed.
81	ProcessCode	Y	Can be either T(Tagged) or N/No Tag) T - Order is to be tagged when submitted to the host N - Order is not tagged when submitted to the host
18	ExecInst	Y	Can be Either R(Retained) or P(Purged) For Details, refer to section “4.7 Order Behaviour by ExecInst type”
5030	Shared	Y	Can be either S(Shared) or N(Not Shared) S- Order is shared (i.e. can be viewed, modified, cancelled) by all traders/operators in the Firm designated with Shared rights for the Share Group specified in Tag 5029 N – Order is not shared (i.e. can be viewed, modified, cancelled) only by the owner of the order.
5029	Shared Group Id.	N	Required if 5030=S. Shared group id.
58	Text	Y	Six Character Comment Field
5038	NoLegs	Y	Number of repeating legs to follow (>=2 and <=6)
55	Symbol	Y	Contract Identifier (excludes spreads)
44	Price	Y	Order leg price
5040	LegVolumeRatio	Y	Custom leg volume ratio
5041	LegSide	Y	Custom leg side B=buy or S=sell
126	ExpireTime	Y	Required if OrderType = 2
	Standard Trailer	Y	

Note: The Legs included in a Custom Market order are required to be sorted as follows:

- 1) Alphabetically i.e. Example 1: **APZ2** should appear before **IRU2**. Example 2: **XTZ295800C** should appear before **YTZ295600P**. Example 3: **USG332500P** should appear before **VWF3**
- 2) Then chronologically i.e. Example: **APZ2** should appear before **APZ3**
- 3) Then numerically i.e. By the lowest strike then the highest strike and then by the Call indicator and then by the Put indicator i.e. Example 1: **APZ232000C** should be before **APZ232500C**,
Example 2: **APZ232000C** should be before **APZ232000P**.

5.4.22 Execution Reports

The execution report message is used to:

1. Confirm the receipt of an order
2. Confirm changes to an existing order
3. Relay fill information as orders are traded
4. Reject orders
5. Confirm changes to a trade -

Each execution message will contain amongst other fields:

- API transaction reference (ClOrdID) - for tracking of the original request (set to 0 (zero) for all non-owner shared orders/trades). **NOTE:** See Tag 11 definition for uniqueness requirements.
- ASX Trade24 order number (OrderID) – unique reference for an order
- Order/Trade Status
- Order/Trade details (Side/Symbol/Account/Price etc.)

Execution report messages can be transmitted as transaction types (ExecTransType) NEW, CANCEL, UPDATE.

- The NEW transaction type indicates that this message represents a new order, a change in status of the order, or a new fill against an existing order. The combination of the ExecTransType and OrdStatus fields will indicate how the message is to be applied to an order.
- The CANCEL transaction type applies at the execution level. The Cancel transaction will be used to cancel an order.
- The UPDATE transaction type applies at the execution level and is used to update an order or trade.

NOTE: The cancelled and updated order status is sent in response to accepted cancel/replace requests. These requests are only acted upon when there is an outstanding order quantity. Requests to replace OrderQty to a level less than the CumQty will result in the cancellation of the remaining volume. The updated trade status values are set in response to accepted trade update requests.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 8
37	OrderID	Y	Order Number assigned by ASX Trade24, Always Present
11	ClOrdID	Y	The originating API will use this field to locate all outstanding order messages waiting for acceptance/rejection from the Gateway. Always must be present. In the case of a shared order, this field will contain a value of 0 when sent to non-originating API. This indicates it is a shared order which is currently owned by another API. In trade message this field will be the original API generated Order Number for all non-shared orders and 0 (zero) for shared orders. NOTE: See Tag 11 definition for uniqueness requirements
17	ExecID	Y	Deal number Assigned By the host

20	ExecTransType	Y	Transaction Type New/Cancel/Correct/Download, Always Present
39	OrdStatus	Y	Status of Submitted Order/Update/Cancel - New(Accepted)/Rejected/Cancelled/Filled/Partially Filled/Expired or Status of a modified trade - Modified/ Always Present
103	OrdRejReason	Y	Only Present when OrdStatus = 8 (Rejected), only in response to an invalid order message
1	Account	Y	Account Number, Present for Order & Trade Messages.
55	Symbol	Y	Instrument/Contract Identifier, Order & Trade Messages Note: <ul style="list-style-type: none"> • For Order Confirm for Spreads, the spread contract Symbol is used. • For Trade Confirm for Spreads, the underlying contract symbol is used.
54	Side	Y	BID/ASK Indicator, Order & Trades Only
38	OrderQty	Y	Original or modified Order Volume, Order and Trade Messages
44	Price	Y	Order/Trade Price, Order and Trade Messages.
32	LastShares	Y	Volume Traded, Only Present For Trades Note: For spread trades, this value is the order volume multiplied by the spread ratio (i.e. the volume traded in the underlying contract)
100	ExDestination	Y	Exchange ID. Order & Trade Messages.
14	CumQty	Y	Total Volume Traded, cumulative volume of all partial trades for a particular order.
60	TransactTime	Y	Time Transaction Occurred/accepted by the host.
81	ProcessCode	Y	Can be either T(Tagged) or N(No Tag) T - Order is to be tagged when submitted to the host N - Order is not tagged when submitted to the host For orders only.
18	ExecInst	Y	Can be Either R(Retained) or P(Purged) Present for Orders only. For Details, refer to section "4.7 Order Behaviour by ExecInst type"
5030	Shared	Y	Can be either S(Shared) or N(Not Shared) S- Order is shared (i.e. can be viewed, modified, cancelled) by all traders/operators in the Firm designated with Shared rights for the Share Group specified in Tag 5029 N – Order is not shared (i.e. can be viewed, modified, cancelled) only by the owner of the order.
5029	Shared Group Id.	N	Required if 5030=S. Shared group id.
5059	Shared Trader ID ²⁵	N	Current Owner of the shared trade. Required if 5030 = S and OrdStatus = 1, 2, M, m, X or x

58	Text	Y	Only Present when OrdStatus = 8(rejected). Order/Update/Cancellation Error Message. Six Character comment code present for orders and present for modified trade if Text was modified.
40	OrdType	Y	Not present if OrdStatus = 1, 2, M, m, X or x (trade messages)
126	ExpireTime	Y	Present if OrderType = 2
	Standard Trailer	Y	

5.4.23 Custom Order Execution Report

The custom order execution report is used to:

1. Confirm the receipt of a custom order
2. Confirm changes to an existing custom order
3. Reject custom orders

Attributes/Behaviour of custom order execution reports are identical to execution reports described in the execution report section.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 20 – Execution Report Custom Order
37	OrderID	Y	ASX Trade24 order number, Always Present
11	ClOrdID	Y	The originating API will use this field to locate all outstanding order messages waiting for acceptance/rejection from the Gateway. Always must be present. In the case of a shared order, this field will contain a value of 0 when sent to non-originating API. This indicates it is a shared order is currently owned by another API. In trade message this field will be the original API generated Order Number for all non-shared orders and 0 (zero) for shared orders. NOTE: See Tag 11 definition for uniqueness requirements
20	ExecTransType	Y	Transaction Type New/Cancel/Correct/Download, Always Present
39	OrdStatus	Y	Status of Submitted Order/Update/Cancel - New(Accepted)/Rejected/Cancelled/Expired Always Present
103	OrdRejReason	Y	Only Present when OrdStatus = 8 (Rejected), only in response to an invalid order message.
1	Account	Y	Account Number
100	ExDestination	Y	Exchange ID.
38	OrderQty	Y	Original Order Volume

14	CumQty	Y	Total Volume Traded, cumulative volume of all partial trades for a Particular order.
60	TransactTime	Y	Time Transaction Occurred/accepted by the host.
81	ProcessCode	Y	Can be either T(Tagged) or N(No Tag) T - Order is to be tagged when submitted to the host N - Order is not tagged when submitted to the host
18	ExecInst	Y	Can be Either R(Retained) or P(Purged) For Details, refer to section "4.7 Order Behaviour by ExecInst type"
58	Text	Y	Only Present when OrdStatus = 8(rejected). Order/Update/Cancellation Error Message. Six Character comment code for orders.
5030	Shared	Y	Can be either S(Shared) or N(Not Shared) S- Order is shared (i.e. can be viewed, modified, cancelled) by all traders/operators in the Firm designated with Shared rights for the Share Group specified in Tag 5029. N – Order is not shared (i.e. can be viewed, modified, cancelled) only by the owner of the order.
5029	Shared Group Id.	N	Required if 5030=S. Shared group id.
40	OrdType	Y	Not present if OrdStatus = 1 or 2 (trade messages)
5038	NoLegs	Y	Number of repeating legs to follow (>=2 and <= 6)
55	Symbol	Y	Valid custom market leg contract identifier (excludes spreads)
44	Price	Y	Order leg price
5040	LegVolumeRatio	Y	Custom leg volume ratio
5041	LegSide	Y	Custom leg side B=buy or S=sell
126	ExpireTime	Y	Present only if OrderType = 2)
	Standard Trailer	Y	

5.4.24 Update Request

- The Update request is used to UPDATE an existing order and should not be used to cancel an outstanding order.
- Update requests which cannot be processed will be rejected using the Cancel Reject message; with the ClOrdID of the Cancel/Replace message inserted in the ClOrdID field of the Cancel Reject message for tracking.
- Update requests which can be processed will be accepted using the Execution message. To maintain data integrity, the Gateway will return all fields of the modified order to the API.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = G
11	ClOrdID	Y	Unique identifier of Cancel/Replace request as assigned by the API. Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the modification request is rejected or the ClOrdID field of the Execution report if the modification request is accepted.
37	OrderID	Y	ASX Trade24 Order Number
55	Symbol	Y	Must match original order (not present on Custom Order Updates)
54	Side	Y	Must match original side (not present on Custom Order Updates)
1	Account	Y	Only Present if the Account number is to be modified. The New Account
38	OrderQty	Y	Only Present if the Quantity is to be changed
40	OrdType	Y	Only Present if the Order Type is to be changed
126	ExpireTime	N	Only Present if OrderType is present and OrderType = 2
81	ProcessCode	Y	Only Present If Tagged field has been modified Can be either T(Tagged) or N/No Tag) T - Order is to be tagged when submitted to the host N - Order is not tagged when submitted to the host
18	Execlnst	Y	Only present if Retained/Purge Field is modified Can be Either R(Retained) or P(Purged) For Details, refer to section "4.7 Order Behaviour by Execlnst type"
58	Text	Y	Only Present if the comment is modified
44	Price	Y	Only Present if the Price has been modified (not present on Custom Order Updates)
	Standard Trailer	Y	

5.4.25 Order Cancel Request

- The order cancel request message is used to request the cancellation of an existing order.
- The CxlType field is used to define the cancel request as either standard (one order) or mass (one or more orders).
- A cancel request is assigned a unique transaction id in the ClOrdID field. If rejected/accepted, the ClOrdID of the cancel request will be sent in the ClOrdID field of the resulting execute report.
- When performing a cancel/mass cancel the Gateway will return the ASX Trade24 order number(s) in the OrderID field of the resulting execute message. To maintain data integrity between the API and Gateway, the API should wait for the cancellation execution report(s) prior to removing orders from trade book.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = F
37	OrderID	Y	ASX Trade24 order number Not required for mass cancels
11	ClOrdID	Y	Unique identifier of cancellation request as assigned by the API. . Note that this identifier will be used in ClOrdID field of the Cancel Reject message if the modification request is rejected or the ClOrdID field of the Execution Message if the modification request is accepted.
125	CxlType	Y	Type of cancel to perform
1	Account	Y	Only Present When cancelling all orders for an account
55	Symbol	Y	Only Present when doing a mass cancel on an Instrument or an Instrument-Bid/Ask combination.
54	Side	Y	Only Present when Doing a mass cancel on Bid/Ask or an Instrument-Bid/Ask Combination.
	Standard Trailer	Y	

5.4.26 Order Cancel Reject

- The order cancel reject message is sent when a cancel request or update request message cannot be honored.
- Requests to change price or decrease quantity are executed if an outstanding quantity exists. Orders which are filled cannot be changed.
- When rejecting an Update Request, the ClOrdID of the request message is inserted in the ClOrdID field of the Cancel Reject message for identification.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 9
37	OrderID	Y	ASX Trade24 order number
11	ClOrdID	Y	Unique transaction identifier assigned to the cancel request by the API
102	CxlRejReason	Y	Error Number
58	Text	Y	Text Error Message
	Standard Trailer	Y	

5.4.27 Trade Update Request

The trade update request is used to modify details of a trade executed either by the API or as part of a valid shared order group. The API will be able to modify the Account and/or Comment fields of the trade.

Upon receipt of this message, the Gateway will either send an Execution Report containing the appropriately updated fields and tag values, or a User Defined Request Reject message in the case of an error.

In the case of an error, the ReferenceNo of the original Trade Update Request message will be sent in the User Defined Request Reject message. Edits on Account and Comment will be similar to those implemented for order modification.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = U
5001	UserMsgCat	Y	Set To 21 For Trade Update Request
5036	ReferenceNo	Y	Transaction Reference Number
17	ExecID	Y	Deal Number
54	Side	Y	Buy/Sell Indicator
55	Symbol	Y	Contract Identifier
1	Account	N	Should only be present if Account value is being modified.
58	Text	N	Should only be present if Comment value is being modified.
	Standard Trailer	Y	

6 Field Definitions

The following is a catalog of fields used to define the application and session protocol messages.

Field ID (TAG)	Field Name	Format	Description
1	Account	char	Account Number as defined as <Account><CTI><House/Client Indicator>. Account is 1-10 alphanumeric characters, CTI can be 1 or 2 and House/Client can be H or C e.g. 1H or 2C.
7	BeginSeqNo	int	Message sequence number of first record in range to be resent
8	BeginString	char	Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. Valid values = FIX.4.0
9	BodyLength	int	Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. Valid values = 0 – 9999
10	CheckSum	char	Three byte, simple checksum (see Appendix A for description). ALWAYS LAST FIELD IN RECORD; i.e. serves, with the trailing <SOH>, as the end-of-record delimiter. Always defined as three characters.
11	ClOrdID	char	Unique identifier for transactions, assigned by the API. Whenever an Order, Cancel or Update is submitted this value should be incremented by 1 (one). The first Transaction that is submitted to the Gateway should have a ClOrdID = 1. Uniqueness must be guaranteed. Will be used by the API to locate the original request. The ClOrdID will range from 1 – 9999999. A shared order sent to a non-originating API will have a ClOrdID = 0 (zero) in the Execution Report. NOTE: Any ClOrdID values received from a Trade/Order download must not be reused. Failure to do so will result in unpredictable and/or incorrect execution reports for the ClOrdID values that are duplicated. Uniqueness of the ClOrdID field must be guaranteed.
14	CumQty	int	Current volume traded, for an order. This field will be the cumulative volume of all full/partial trades, for an order. For Example: If order 1 was submitted with a volume of 100 and there were three hits (partial trades), against order 1, of volumes 1, 3, 5. This field would be set to 9 (1 + 3 + 5).

16	EndSeqNo	int	<p>Message sequence number of last record in range to be resent. If request is for a single record BeginSeqNo = EndSeqNo.</p> <p>If request is for all messages subsequent to a particular message, EndSeqNo = "9999999"</p>
17	ExecID	int	<p>ASX Trade24 Deal Number. Unique by Trading Day.</p> <p>Valid Values: 1-999999999</p>
18	ExecInst	char	<p>Instructions for handling an order if the Gateway disconnects from the Host, can be one of two values P/R.</p> <p>P = Purge this order</p> <p>R = Retain this order</p> <p>For further details, refer to section "4.7 Order Behaviour by ExecInst type"</p>
20	ExecTransType	char	<p>Identifies transaction type</p> <p>Valid values:</p> <p>0 = New</p> <p>1 = Cancel</p> <p>2 = Correct</p> <p>D = Download</p>
32	LastShares	int	Volume bought/sold on this fill.
34	MsgSeqNum	int	<p>Integer message sequence number.</p> <p>Valid values = 0 – 9999999</p>
35	MsgType	char	<p>Defines message type. ALWAYS THIRD FIELD IN MESSAGE.</p> <p>Note: A "U" as the first character in the MsgType field indicates that the message format is privately defined between the sender and receiver.</p> <p>Valid values:</p> <p>0 = Heartbeat</p> <p>1 = Test Request</p> <p>2 = Resend Request</p> <p>3 = Reject</p> <p>4 = Sequence Reset</p> <p>5 = Logout</p> <p>8 = Execution Report</p> <p>9 = Order Cancel Reject</p> <p>A = Logon</p> <p>D = Order - Single</p> <p>F = Order Cancel Request</p> <p>G = Order Cancel/Replace Request (Update)</p> <p>U = User Defined Message</p>

36	NewSeqNo	int	New Sequence Number Valid values = 0 – 9999999
37	OrderID	char	Unique identifier for Order as assigned by ASX Trade24. Uniqueness will be guaranteed within a Trading Week, (Monday-Friday). Active GTC orders are also guaranteed to be unique. Valid Values = 1 – 9999999.
38	OrderQty	int	Volume ordered. Valid values = 0 – 99999
39	OrdStatus	char	Identifies current status of order. Valid values: 0 = New 1 = Partially filled 2 = Filled 4 = Cancelled By user 5 = Replaced 8 = Rejected 9 = Purged By System C = Expired(Fill or Kill has expired) M = Modified Trade m = Modified Partial X = Cancelled Trade x = Cancelled Partial Trade
40	OrdType	char	Order type. Valid values: 1 = Limit Order (LIM) 2 = Timed Order (TIM) 3 = Fill or Kill (FOK) 4 = Market (MKT) 5 = Market If Touched (MIT) 6 = Stop Order (STP) 7 = Stop Limit (STM) 8 = Good Till Cancel (GTC) 9 = Discretionary (DSC) 10=Market Limit (MLM) Order Types 1, 2, 3, 8 and 10 are execution order types. The remaining types are “memo” order types, and behave similar to LIM orders.
43	PossDupFlag	char	Indicates possible retransmission of message with this sequence number Valid values: Y = Possible duplicate N = Original transmission

44	Price	float	Order/Trade price. Valid values = 0 - 99999999
45	RefSeqNum	int	Reference message sequence number, of rejected/garbled messages. These are rejected FIX messages and not rejected order or cancel messages. Valid values = 0 - 9999999
49	SenderCompID	char	Assigned value used to identify firm sending message.
50	SenderSubID	char	Assigned value used to identify specific message originator (Trader ID)
52	SendingTime	time	Time of message transmission (always expressed in GMT)
54	Side	char	Side of order Valid values: 1 = Buy 2 = Sell
55	Symbol	char	Commodity/Contract/Instrument identifier. When a message specifies a commodity code, the symbol field will contain the two character commodity code. When the message specifies a contract or instrument the symbol field will contain the CCMY/CCMYSSSSSP/Co tradable instrument as entered into the system. The 'o' is used to indicate that the contract is a Single Session option. When the message specifies an Intra Spread contract, the symbol field will contain CCMYMY, where MY represents the month/year of the 1 st and 2 nd legs of the spread. When the message specifies an Inter Spread contract, the symbol field will contain CCMYCCMYab where CCMY represents the commodity code/month/year of the 1 st and 2 nd legs of the spread and <u>ab</u> represents the volume ratio between the 2 legs, e.g. An inter-spread contract with a 10:31 volume ratio would be represented as CCMYCCMY1031. Note: The Option strike commodity code contains 5 characters (CCMYSSSSSP/Co.
58	Text	char	Free format text string
60	TransactTime	time	Time of execution/order creation (expressed in GMT)
81	ProcessCode	char	Defines the order as T (Tagged) or N (No Tag). This provides the ability to group a set of orders and mass cancel them.

95	RawDataLength	int	Number of bytes in raw data field.
96	RawData	Data	Unformatted raw data. Note: may contain the <SOH> character
100	ExDestination	char	Execution destination. Which exchange the order is to be submitted to. In the case of a trade, what exchange the trade was executed on. Valid Values: "SFE" , "CFD" and "NZFOE" (without the quotes)
102	CxlRejReason	int	Code to identify reason for cancel rejection. The Gateway will pass back both the code and the error text. Definitions have been updated to reflect exact text passed back from the Gateway. Valid values: 0 = Contract not trading 1 = Order not found 5 = Invalid volume 6 = Invalid account 7 = Invalid order type 8 = Invalid ProcessCode 9 = Invalid Execlnst 10 = Invalid contract 11 = Invalid Side 12 = Outside trading limits 13 = Price outside market 15 = Invalid request 16 = Invalid Custom ratio
103	OrdRejReason	int	Code to identify reason for order rejection. The Gateway will pass back both the code and the error text. Definitions have been updated to reflect exact text passed back from the Gateway. Valid values: 1 = Invalid contract 2 = Contract not trading 3 = Outside trading limits 5 = Invalid volume 6 = Invalid account- 7 = Invalid order type 8 = Invalid ProcessCode 9 = Invalid Execlnst 10 = Price outside market 11 = Invalid Side 12 = Invalid shared order indicator 15 = Invalid order 16 = Invalid Custom ratio 17 = Invalid Custom Leg Sequence 18 = Price Outside MIR Limits

108	HeartBtInt	int	Heartbeat interval (seconds), set to 1 second
112	TestReqID	char	Identifier included in Test Request message to be returned in resulting Heartbeat Format: HHMMSS
122	OrigSendingTime	time	Original time of message transmission (always expressed in GMT) when transmitting orders as the result of a resend request.
123	GapFillFlag	char	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent. Valid values: Y = Gap Fill message, MsgSeqNum field valid N = Sequence Reset, ignore MsgSeqNum
125	CxlType	char	Defines if cancel is for part or all of the remaining quantity of an order. Valid values: F- Standard, cancel remaining volume for an order. 1 - All Orders for MFOS 2 - All Tagged Orders for MFOS 3 - All Bid Orders 4 - All Ask Orders 5 - All Commodity/Contract Orders 6 - All Commodity/Contract Bid Orders 7 - All Commodity/Contract Ask Orders 8 - All orders for Account
126	ExpireTime	time	Time/Date of order expiration (always expressed in GMT)

5001	UserMsgCat	int	<ul style="list-style-type: none"> 0 – Download Commodity Request 1 – Commodity Download 2 – Contract Parameter Download Request 3 – Contract Parameter Download 4 - Market Download Request 5 - Market Download 6 - Market Update 7 – Trade/Order Download 8 – Trade/Order Download Complete 9 – User-Defined Request Reject 10 - Filtered Market Update Request 11 - Filtered Market Download Request 12 - At-Best Request 13 - At-Best Update 14 - DOM Request 15 - DOM Update 16 – User Text Message 17 – Request For Quote 18 – Custom Market Update 19 – New Custom Order 20 – Custom Order Execution Report 21 - Trade Update Request
5002	InstrmntIndicator	char	<p>Flag which defines contents of the Symbol field.</p> <ul style="list-style-type: none"> A - Inter Spreads C - Intra Spreads F – Future O – Option
5003	LastMessage	int	<p>Last Message In a Download</p> <ul style="list-style-type: none"> 0 – Not last Message 1 – Last Message
5004	Currency	char	Currency of the listed contract
5005	TickIncrement	float	Futures/Options Price Tick Increment

5006	SessionNo	int	Session No Valid values: 1-100 (Not actively used by ASX Trade24 but a value between the above range must be set to avoid errors. Please refer to section 5 for more information)
5009	MonthYear	char	MY – Month Year combination for an instrument
5010	UnderLyingFut	char	Underlying Futures for an Instrument (i.e. serials)
5011	LastTradedVolume	int	Volume of the last trade
5012	ChangeStatusOf	char	Change the status of the following: C - The Commodity and all its instruments F – Future(s) within a commodity O – Option(s) within a commodity B - Both Futures and Options within a commodity M - Market (used only when the entire system has been temporarily halted (i.e. in conjunction with 5014 = Suspended) or resumed (i.e. in conjunction with 5014 = Opened) N – Single Session Option Series (used to distinguish overnight and intra-day options from regular options at a series level)
5013	OptionsStrikeP/C	char	Options strike price (6 characters - CCMYSSSSSP/Co), Put/Call Indicator, single session option indicator P – Puts C – Calls Po – Puts (single session options) Co – Calls (single session options)

5014	MarketStatus	char	The status of an instrument p – Pending (unavailable for trading) P – Pre-opened I – Levelling O – Opened d – Pre-Price Discovery (similar to Pending) D – Price Discovery (similar to Pre-Open) H – Halted S – Suspended C – Closed I – Inactive (applies to option strikes only) i – Settling (Interim) F – Settled (Final) L – Locked U – Unavailable R – Regulatory Halt (similar to Pre-Open)
5015	LowTradePrice	float	The lowest Price the Contract has traded for the day/session
5016	HighTradePrice	float	The highest Price the Contract has Traded for the day/session
5017	SettlementPrice	float	Contracts Settlement Price
5018	Volatility	float	An Options Series' Volatility
5019	OpenPrice	float	Open Price of the Contract for the day/session
5020	ClosePrice	float	Closing Price for the Contract
5021	FOBIndicator	char	Used to indicate that an instrument has Futures, Options or Both F – Futures only O – Options only B – Both Futures and Options
5022	AskPrice	float	The best ask price in the market
5023	AskVolume	int	The volume at the best ask price

5024	BidPrice	float	The Best Bid Price In the market
5025	BidVolume	int	The volume at the best bid price
5026	LastTradedPrice	float	The price of the last trade
5028	InterSprdRatio	char	Volume ratio for Legs of an Inter Instrument spread (i.e. X:Y)
5029	Shared Group Id.	int	1 - 50
5030	Shared	char	Can be either S (Shared) or N (Not Shared) S – Order is shared (i.e. can be viewed, modified, cancelled) by all Gateways/APIs in a firm, if their Gateway is a member of the shared order group specified. N – Order is not shared (i.e. can be viewed, modified, cancelled only by the owner of the order).
5031	LastBuyer	char	Last buying firm. Blank due to anonymous market.
5032	LastSeller	char	Last selling firm. Blank due to anonymous market.
5033	FilterType	char	0 = All 1 = By Exchange 2 = By Instrument 3 = By Custom Market
5034	InterSpreadsFlag	char	Include Inter-Instrument Spreads in filter for market update or market download Valid values: Y = include N = do not include
5035	Enable	char	Valid values: Y = enable N = disable
5036	ReferenceNo	int	Unique reference number, Valid values = 1 – 9999999

5037	CustomVolume	int	Custom volume
5038	NoLegs	int	Number of repeating custom legs, (>=2 and <=6)
5039	LegPrice	int	Custom leg price
5040	LegVolumeRatio	int	Custom leg volume ratio
5041	LegSide	char	Custom leg side B=buy or S=sell
5042	NoAskLevels	int	Number of repeating ask levels
5043	AskFirm	char	Asking firm identifier. Blank due to anonymous market. Where 5043 = *** indicate the order is implied
5044	AskVol	int	Asking volume
5045	NoBidLevels	int	Number of repeating bid levels
5046	BidFirm	char	Bidding firm identifier. Blank due to anonymous market. Where 5046 = *** indicate the order is implied
5047	BidVol	int	Bidding volume
5048	SendMsgType	char	0 – UserId (To/From) 1 – ASX 24 Operations (To/From) 2 – All Users (To)
5049	SendMsgUserId	char	6 character UserId field
5050	SendMsgText	char	60 character message text
5051	QuoteVolume	int	Request for quote volume
5052	QuoteFirm	char	Quoting firm identifier

5053	UserMsgCatReq	int	UserMsgCat for user-defined requests supporting rejection 10 - Filtered Market Update Request 11 - Filtered Market Download Request
5054	FractionalIndicator	char	Contract fractional indicator.
5055	TotalTradedVolume	int	Total Traded volume for a contract
5056	InterimSettlement	int	Contract Interim Settlement.
5057	ImpliedPrice	int	Implied Price Indicator 1 – Implied Bid Price 2 – Implied Ask Price 3 – Bid and Ask are implied prices
5059	Shared Trader ID	char	Current Owner of a shared trade

7 APPENDIX A - CheckSum Calculation

The checksum of a FIX message is calculated by summing every byte of the message up to but not including the checksum field itself. This checksum is then transformed into a modulo 256 number for transmission and comparison. The checksum is calculated after all encryption is completed, i.e. the message as transmitted between parties is processed.

For transmission, the checksum must be sent as printable characters, so the checksum is transformed into three ASCII digits.

For example, if the checksum has been calculated to be 274 then the modulo 256 value is 18. This value would be transmitted as |10=018| where "10=" is the tag for the checksum field.

A sample code fragment to generate the checksum field is as follows:

```
char *GenerateChecksum( char *buf, long bufLen )
{
    static char tmpBuf[ 4 ];
    long idx;
    unsigned int cks;

    for( idx = 0L, cks = 0; idx < bufLen; cks += (unsigned int)buf[ idx++ ] );
    sprintf( tmpBuf, "%03d", (unsigned int)( cks % 256 ) );
    return( tmpBuf );
}
```

8 Field Index Sorted by Tag Number

1	Account	5001	UserMsgCat
7	BeginSeqNo	5002	InstrmntIndicator
8	BeginString	5003	LastMessage
9	BodyLength	5004	Currency
10	Checksum	5005	TickIncrement
11	ClOrdID	5006	SessionNo
14	CumQty	5009	MonthYear
16	EndSeqNo	5010	UnderLyingFut
17	ExecID	5011	LastTradedVolume
18	ExecInst	5012	ChangeStatusOf
20	ExecTransType	5013	OptionsStrikeP/C
32	LastShares	5014	MarketStatus
34	MsgSeqNum	5015	LowTradePrice
35	MsgType	5016	HighTradePrice
36	NewSeqNo	5017	SettlementPrice
37	OrderID	5018	Volatility
38	OrderQty	5019	OpenPrice
39	OrdStatus	5020	ClosePrice
40	OrdType	5021	FOBIndicator
43	PossDupFlag	5022	AskPrice
44	Price	5023	AskVolume
45	RefSeqNum	5024	BidPrice
49	SenderCompID	5025	BidVolume
50	SenderSubID	5026	LastTradedPrice
52	SendingTime	5027	Not Used at this time
54	Side	5028	InterSprdRatio
55	Symbol	5029	Shared Group Id.
58	Text	5030	Shared
60	TransactTime	5031	LastBuyer
81	ProcessCode	5032	LastSeller
95	RawDataLength	5033	FilterType
96	RawData	5034	InterSpreadsFlag
100	ExDestination	5035	Enable
102	CxlRejReason	5036	ReferenceNo
103	OrdRejReason	5037	CustomVolume
108	HeartBtInt	5038	NoLegs
112	TestReqID	5039	LegPrice
122	OrigSendingTime	5040	LegVolumeRatio
123	GapFillFlag	5041	LegSide
125	CxlType	5042	NoAskLevels
126	ExpireTime	5043	AskFirm

5044 AskVol	5053 UserMsgCatReq
5045 NoBidLevels	5054 FractionalIndicator
5046 BidFirm	5055 TotalTradedVolume
5047 BidVol	5056 InterimSettlement
5048 SendMsgType	5057 ImpliedPrice ²⁵
5049 SendMsgUserId	5058 Reserved
5050 SendMsgText	5059 Shared Trader ID ²⁵
5051 QuoteVolume	
5052 QuoteFirm	

9 Messages not Implemented

Advertisements

Indication of Interest

News

Email

Quote Request

Order Status Request

Allocation

Allocation ACK

New Order List

List Status

List Execute

List Cancel Request

List Status Request

10 Unused Fields Sorted by Field Tags

2	AdvID	79	AllocAccount
3	AdvRefID	80	AllocShares
4	AdvSide	82	NoRpts
5	AdvTransType	83	RptSeq
6	AvgPx	84	CxlQty
12	Commission	85	NoDlvyInst
13	CommType	86	DivvyInst
15	Currency	87	AllocStatus
19	ExecRefID	88	AllocRejCode
21	HandlInst	89	Signature
22	IDSource	90	SecureDataLen
23	IOId	91	SecureData
24	IOIOthSvc	92	BrokerOfCredit
25	IOIQltyInd	93	SignatureLength
26	IOIRefID	94	EmailType
27	IOIShares	97	PossResend
28	IOITransType	98	EncryptMethod
29	LastCapacity	99	StopPx
30	LastMkt	104	IOIQualifier
31	LastPx	105	WaveNo
33	LinesOfText	106	Issuer
41	OrigClOrdID	107	SecurityDesc
42	OrigTime	109	ClientID
46	RelatdSym	110	MinQty
47	Rule80A	111	MaxFloor
48	SecurityID	113	ReportToExch
51	SendingDate	114	LocateReqd
53	Shares	115	OnBehalfOfID
56	TargetCompID	116	OnBehalfOfSubID
57	TargetSubID	117	QuoteID
59	TimeInForce	118	NetMoney
61	Urgency	119	SettlCurrAmt
62	ValidUntilTime	120	SettlCurrency
63	SettlmntTyp	121	ForexReq
64	FutSettDate	124	NoExecs
65	SymbolSfx	127-211	Various Fields
66	ListID		
67	ListSeqNo		
68	ListNoOrds		
69	ListExeclnst		
70	AllocID		
71	AllocTransType		
72	RefAllocID		
73	NoOrders		
74	AvgPrxPrecision		
75	TradeDate		
76	ExecBroker		
77	OpenClose		
78	NoAllocs		