ASX 24 FIX Drop Copy Specification

Version 2.05 | October 2023

Table of Contents

1. FIX Drop Copy Introduction

ASX provides the facility to receive copies of outbound messages pertaining to a participant's order and execution flow using the FIX protocol. Drop copies can be used to produce a consolidated view of order status for a configured set of users. A Trading participant may require drop copies of all executions for trading users within the same firm, or another firm under agreement. Drop copy can be particularly useful for clearing members wanting to consolidate all trades for clients that the Clearing firm is responsible for.

This document provides guidance on using FIX protocol messages to:

- Establish and maintain a FIX session
- Receive information about orders and trades
- Request for trades (fills) for a specified date range

These rules of engagement are based on the FIX 5.0 SP2 specification with extension pack 196 and best practice guidelines as published by the FIX Trading Community. Unless specifically stated, field numbers, names, and data types are as published by the FIX specification. A full explanation of the FIX protocol is out of scope of the document therefore customers should refer to FIX Trading Community for a full understanding of the protocol, prior to using this guide.

Several other aspects of operating the services need to be taken into consideration in effectively utilising the drop copy service, such as:

- Support for drop copy users
- Pre-trade risk management
- Trading protection limits
- Failover

Details on these aspects can be found on the ASX website and from your Technical Account Manager.

1.1. Supported Versions

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.

ASX Gateways support FIX version 5.0 SP2 EP 196.

A detailed description of FIX is available from http://www.fixtradingcommunity.org and includes all the technical specifications.

1.2. Document Information

This document describes:

- Common message structures, including standard header and standard trailer definitions
- FIX session layer—how FIX sessions are established and maintained
- Business layer introduction—general aspects of the business layer of the protocol
- Drop copy messages—valid drop copy messages.

Please refer to the ASX FIX Drop Copy Specification Updates document to see changes between the V1.01 and V1.02 specifications.

1.2.1. Terms and Acronyms

Term / Acronym	Description
FIX	Financial Information eXchange Protocol
FPL	FIX Protocol Limited
ISIN	International Securities Identification Number. Unique identifier issued to identify each financial instrument.
UTC	Coordinated Universal Time is a high-precision, atomic time standard often referred to as GMT.
CFI	Classification of Financial Instruments

1.2.2. Usage of String Datatypes

String datatype field definitions, that do not have specified values or lengths in FIX, have been updated to display field lengths, in square brackets, supported by ASX. For example, String [19] indicates that 19 characters is the maximum number of characters that ASX will send or process on incoming messages.

In cases where the data length is dependent on other parameters, the phrase *see commen*t appears instead, indicating that additional details are available in the Comment column. For example, the PartyID (448) datatype is specified as *String [see comment]* because it depends on the PartyRole (452) being transmitted.

1.2.3. Usage of UTCTimestamp

ASX 24 supports the UTCTimestamp second (YYYYMMDD-HH:MM:SS) and milliseconds (YYYYMMDD-HH:MM:SS.sss) formats. Times with higher precision or that do not meet those formats will be rejected.

1.2.4. Handling of unsupported messages and tags

Any message that is not listed in this specification will be rejected.

If a message listed in this specification is received with a tag that is not in the specification, the message will be rejected.

1.2.5. Definition of required column values

For messages transmitted to ASX

Value	Definition
Y	Defined as required in FIX 5.0 SP2 EP 196.
C	Conditionally required by either FIX protocol or by ASX to implement functionality.
Blank	Tag may be omitted.

For messages transmitted from ASX

Value	Definition
Y	Defined as required in FIX 5.0 SP2 EP 196.
Blank	Always provided by ASX unless text specifies otherwise.

1.3. Version History

This document has been revised according to the table below:

Version	Date	Comment
V1.0	July 2016	Final release of specification.
V1.01	August 2016	Added documentation errata and changes per delta document.
V1.02	September 2016	Corrective Release updates – details are included in the Updates document.

Version	Date	Comment
V1.03	December 2016	Added documentation errata and changes per delta document.
V1.04	January 2017	Added additional information surrounding Cancel on Disconnect
V1.05	April 2017	Miscellaneous clarifications on the following messages: • ExecutionReport (8) • TradeCaptureRequest (AE)
V2.00	July 2017	Changes for the Service Release • Addition of QuoteRequest (R) message • Removal *PartyRole* of *Contra clearing firm* (452=18)
V2.01	August 2017	• Clarify Logon (A) to request waiting for Logon ack before sending messages. • Clarify meaning of MatchInst (1625) on multileg trades • Detail how unsupported messages and tags are handled
V2.02	December 2017	• Clarified when SecurityExchange (207) is transmitted in ExecutionReport (8) messages.
V2.03	August 2018	• Clarify what UTCTimestamp format ASX 24 accepts
V2.04	December 2022	• Clarify definition of tag 1128 for PartyRiskLimitReport (CM)
V2.05	October 2023	• Updated naming conventions

2. Common Message Structures

2.1. Standard Header

All FIX messages described in this document contain a standard header, which is defined below.

Tag	Name	Data Type	Reqd	Comment
8	BeginString	String	Y	Identifies beginning of new message and protocol version. Always first field in message. Valid values: FIXT.1.1
9	BodyLength	Length	Y	Message length, in bytes, forward to the CheckSum field. Always second field in message.
35	MsgType	String	Y	Defines message type. Always third field in message. See individual messages for value to be used.
49	SenderCompID [64]	String	Y	Identifies the sender of the message.
56	TargetCompID [64]	String	Y	Identifies the receiver of the message.
34	MsgSeqNum	SeqNum	Y	Message sequence number.
43	PossDupFlag	Boolean	C	Required for retransmitted messages. Indicates possible retransmission of message with this sequence number. Valid values: N = Original Transmission Y = Possible duplicate.

Tag	Name	Data Type	Reqd	Comment
97	PossResend	Boolean	C	Supplied when message may be a duplicate of another message sent under a different sequence number. ASX may send PossResend=True (97=Y) messages after a standby FIX engine has been promoted to act as primary. ASX does not support PossResend (97) on incoming messages.
52	SendingTime	UTCTimestamp	Y	Time of transmission in UTC.
122	OrigSendingTime	UTCTimestamp	C	Required for messages sent as a result of a ResendRequest. Original time of message transmission in UTC.
369	LastMsgSeqNumProcessed	SeqNum		The last MsgSeqNum (34) value received by the FIX engine and processed by downstream application, such as trading engine or order routing system. Can be specified on every message sent. Useful for detecting a backlog with a counterparty.

2.2. Standard Trailer

All FIX messages in this document contain a standard trailer, which is defined below.

Tag	Name	Data Type	Reqd	Comment
10	CheckSum	String	Y	Simple checksum (see Volume 2: "Checksum Calculation" for description in FIX Trading Community FIX Session Layer).

Tag	Name	Data Type	Reqd	Comment
				ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. Always unencrypted.

3. FIX Session Layer

3.1. Connecting to the FIX Server Process

Each FIX client must maintain the address and port for the primary and the standby ASX FIX server. If a server cannot be reached, we recommend performing the following steps:

- Attempt to connect to the other server (standby, if primary attempted and primary, if standby attempted)
- Wait five seconds, and then try to connect to the server again.

3.2. Lost connection

When reconnecting, the subscriber should login again to the session using the next transmitted sequence number. In the event that the sequence number(s) are out of sync, the session should be resynchronised using either standard FIX recovery or NextExpectedMsgSeqNum processing, depending on the type of Login (A) message issued.

Messages about activity that occurs when a FIX session is disconnected are queued and are available to be sent upon reconnection. Examples of such messages are fills on trades and cancellation reports on orders marked for Cancel on Disconnect.

3.3. Logon (A)

The logon message is the first message sent by a user and is used to authenticate the FIX session with the exchange. On successful authentication, the response is a Logon (A) message. The user should wait for the confirming Logon (A) message before sending other messages. Messages sent prior to this confirmation may not be processed.

In the event that logon fails, a Logout (5) message will be sent and the TCP/IP session will be terminated under most circumstances. Circumstances when a Logout (5) message is not sent, include an invalid SenderCompID (49) or TargetCompID (56). See the FIX Trading Community FIX Session Layer for circumstances when a Logout (5) message should not be issued.

If a logon attempt fails, the client should attempt no more than 3 times before taking remedial action requested in the Logout message. Remedial action includes changing the password sent on the Logon (A) message, correcting sequence numbers, or contacting ASX regarding account administration.

The password can be changed by specifying the new password in the NewPassword (925) tag.

Recovery is supported using the ResendRequest (2) and NextExpectedMsgSeqNum (789). Please see the FIX Trading Community FIX Session Layer description for full details of these methods of recovery handling.

Note that if ASX responds to a Logon request with a NextExpectedMsgSeqNum (789) value, indicating that messages should be resent (that is, an implied ResendRequest), the requested messages must be resent before sending any other messages. Any other messages sent by the client will be rejected with a BusinessMessageReject (j) message.

The following points should be noted when reconnecting to the secondary (standby) FIX server:

- NextExpectedMsgSeqNum (789) must be specified on the Logon (A) message

- Messages may be sent with PossResend=true (97=Y) for activity that FIX server detects, but cannot ascertain if the message was sent by the previous FIX server.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = A
98	EncryptMethod	int	Y	Method of encryption. ASX does not use FIX encryption, however this field is required by FIX 5.0 SP2. Please set to zero to specify no encryption. Other valid values are accepted, but will not have any effect. Valid values: 0 = None/Other.
108	HeartBtInt	int	Y	Heartbeat interval in seconds. The same value is used by both sides. Specified by the logon initiator and reflected by the acceptor (ASX). ASX supports values between 5 and 60, recommending 30. ASX does not support a value of zero. Any other values may be accepted but are not supported.
141	ResetSeqNumFlag	Boolean		Indicates both sides of a FIX session should reset sequence numbers. Valid values: N = No Y = Yes, reset sequence numbers. Note MsgSeqNum (34) should be set to 1.

Tag	Name	Data Type	Reqd	Comment
789	NextExpectedMsgSeqNum	SeqNum		Next expected MsgSeqNum value to be received. Required when connecting to the secondary (standby) FIX Drop Copy server. Please see the FIX Trading Community FIX Session Protocol description on the use of this tag for recovery.
553	Username	String [64]	C	FIX username.
554	Password	String [128]	C	Password for username.
925	NewPassword	String [128]		Specifies a new password for the FIX Logon. The new password is used for subsequent logons. The new password must meet at least three of the following criteria: • Contain an English upper case character (A-Z) • Contain an English lower case character (a-z) • Contain a Hindu Arabic numeral (0-9) • Contain one or more of the following non-alphanumeric special characters: !@#$%^&*()_+|~-=\`{}[]:";'<>?,./) • Contain any character that is categorized as an alphabetic character but is not uppercase or lowercase; this includes characters from Asian languages. Passwords must be a minimum of 8 characters in length.

Tag	Name	Data Type	Reqd	Comment
				Passwords are valid for 90 days and when reset must be different to the previous 12 passwords used. The account will lock after 6 failed attempts.
1409	SessionStatus	int		Status of the FIX session. Sent by ASX. Ignored if input by client. Valid values: 0 = Session active 1 = Session password changed.
1137	DefaultApplVerID	String	Y	Specifies the service pack release being applied to the message at the session level. Valid values: 9 = FIX50SP2
58	Text	String [see comment]		Free format text string. Up to 128 characters on incoming message. Outgoing messages may exceed 128 characters.
	StandardTrailer		Y	

3.4. Heartbeat (0)

Heartbeat messages are sent by counterparties to indicate that a connection is still active and as a response to TestRequest (1) messages. Behaviour is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 0
112	TestReqID	String [64]		Identifier included in Test Request message to be returned in resulting Heartbeat. Required when the heartbeat is the result of a Test Request message.
	StandardTrailer		Y	

3.5. TestRequest (1)

To verify a connection is active, a TestRequest (1) message is sent to the counterparty. The recipient of the TestRequest responds with a Heartbeat (0) message. Failure to respond to a TestRequest message may trigger a disconnection by the sender.

The behaviour is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 1
112	TestReqID	String [64]	Y	Identifier of this Test Request. To be returned in the Heartbeat generated upon receipt of the Test Request.
	StandardTrailer		Y	

3.6. ResendRequest (2)

The ResendRequest (2) message is sent to request the retransmission of messages. Note that either party may send a ResendRequest.

When the client receives a ResendRequest (2) from the ASX, the requested messages must be resent before sending any other messages. Until the requested messages have been received by ASX, messages sent by the client will be rejected with a BusinessMessageReject (j) message.

The behaviour is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 2
7	BeginSeqNo	SeqNum	Y	Message sequence number of first message in range to be resent.
16	EndSeqNo	SeqNum	Y	Message sequence number of last message in range to be resent. If all messages subsequent to BeginSeqNo are required, set EndSeqNo=0.
	StandardTrailer		Y	

3.7. Reject (3)

The reject message is issued when a message is received but cannot be properly processed due to a session-level, rule violation. As an example, a reject can be issued on receipt of a message with invalid basic data, which successfully passes decryption, check sum, and body length checks. Behaviour is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 3
45	RefSeqNum	SeqNum	Y	MsgSeqNum of rejected message.
371	RefTagID	int		The tag number of the FIX field being referenced.
372	RefMsgType	String		The MsgType (35) of the FIX message being referenced.
373	SessionRejectReason	int		Code to identify reason for a session-level Reject message. Valid values: 0 = Invalid Tag Number 1 = Required Tag Missing 4 = Tag specified without a value 5 = Value is incorrect (out of range) for this tag 6 = Incorrect data format for value 9 = CompID problem 10 = SendingTime accuracy problem 11 = Invalid MsgType 13 = Tag appears more than once. 14 = Tag specified out of required order.

Tag	Name	Data Type	Reqd	Comment
58	Text	String [see comment]		Where possible, message to explain reason for rejection. Up to 128 characters on incoming message. Outgoing messages may exceed 128 characters.
	StandardTrailer		Y	

3.8. SequenceReset (4)

The SequenceReset (4) message is used to inform the counterparty of a new, higher sequence number. This is required in order to skip one or more messages when responding to a ResendRequest (2), or to set a new sequence number after an unrecoverable error.

The behaviour is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 4
123	GapFillFlag	Boolean		Indicates that the Sequence Reset message is replacing administrative or application messages, which will not be resent. Valid values: N = Sequence Reset, ignore MsgSeqNum Y = Gap Fill Message, MsgSeqNum field valid.
36	NewSeqNo	SeqNum	Y	New sequence number.
	StandardTrailer		Y	

3.9. Logout (5)

The logout message is used to initiate or confirm the termination of a FIX session. Logout is normally initiated by the client. The ASX may also initiate a logout, for example, prior to system shutdown.

A Logout message is also used to respond to failed Login (A) requests. On completion of the logout procedure, ASX will close the TCP/IP connection. The logout process followed is as described in the FIX Trading Community FIX Session Layer.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 5
58	Text	String [see comment]		Free format text string. Up to 128 characters on incoming message. Outgoing messages may exceed 128 characters.
1409	SessionStatus	int		Status of the FIX session. Not sent for scheduled server initiated log outs, for example, when the server is shutting down. Valid values: 3 = New session password does not comply with policy 4 = Session logout complete 5 = Invalid username or password 6 = Account locked 7 = Logons are not allowed at this time 8 = Password expired 9 = Received MsgSeqNum(34) is too low 10 = Received NextExpectedMsgSeqNum (789) is too high. 101 = Requested MsgSeqNum unknown (a request has been made for a sequence number that has not been sent by the server).

Tag	Name	Data Type	Reqd	Comment
				103 = MsgSeqNum (34) must be 1 when ResetSeqNumFlag=true (141=Y) 104 = HeartBtInt (108) must be greater than 1 105 = NextExpectedMsgSeqNum (789) must be set after FIX server failover 106 = Logon (A) message could not be recognised 107 = Logon (A) message rejected due to other reasons (see Text (58) for further information) 108 = Unsolicited logout from server (see Text (58) for further information)
	StandardTrailer		Y	

4. Drop Copy Message Details

The following sections cover all supported session messages.

4.1. ExecutionReport (8)

Execution Report (8) messages are used to transmit information about new orders, order modifications, cancellations, and fills. Drop copy execution reports contain the same business information as the execution reports transmitted to the order originator. The following should be noted:

- ClOrdID (11) and OrigClOrdID (41) are as entered on the original connection. For drop copy connections that report the activity of more than one FIX connection, these values may not be unique. OrderID (37), which is supplied by the exchange, is unique and can be used to identify the order.
- Parties block is used to communicate the entities involved, and their roles within the message.
- CopyMsgIndicator is included in all drop copy messages, and is set to Y.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 8
37	OrderID	String [19]	Y	Unique identifier for the order as assigned by ASX. Private to the member firm entering the order. Returned as zero for execution reports notifying of the bust of a leg trade of a spread order.
198	SecondaryOrderID	String [19]		Unique identifier for the order as assigned by ASX. Provided on public market data.
11	ClOrdID	String [128]		Identifier provided by the client.
41	OrigClOrdID	String [128]		Identifier provided by client.

Tag	Name	Data Type	Reqd	Comment
797	CopyMsgIndicator	Boolean		Indicates whether or not this message is a drop copy of another message. Always included in drop copy messages. Valid values: Y = Message is a drop copy.
17	ExecID	String [84]	Y	Unique identifier for the trade assigned by ASX.
19	ExecRefID	String [84]		Execution reference identifier. Supplied when ExecType=Trade Cancel (150=H).
880	TrdMatchID	String [10]		Identifier assigned to a trade by ASX clearing purposes - clearing deal number. For multi-leg product executions, this value is not sent to the clearing system. See individual legs for the clearing deal number for each leg. Provided when ExecType=Trade (150=F).
39	OrdStatus	char	Y	Identifies status of order. Valid values: 0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Cancelled C = Expired

Tag	Name	Data Type	Reqd	Comment
150	ExecType	char	Y	Describes the specific ExecutionReport (8) (e.g. Replaced), while OrdStatus(39) will always identify the current order status (for example, Partially Filled). Valid values: 0 = New 3 = Done for day 4 = Cancelled 5 = Replaced C = Expired D = Restated F = Trade (partial fill or fill) H = Trade Cancel.
378	ExecRestatementReason	int		The reason for restatement. Supplied when ExecType=Restated (150=D). Valid values: 1 = GT renewal / restatement.
1	Account	String [10]		Account supplied on the order.
581	AccountType	int		Type of account associated with an order. Valid values: 1 = Customer 2 = Non-customer.

Tag	Name	Data Type	Reqd	Comment
574	MatchType	String		The point in the matching process at which the trade was matched. Supplied if ExecType=Trade (150=F), ExecType=Trade Cancel (150=H). Also supplied if ExceType=Order Status (150=I) and OrdStatus=Partially Filled (39=1), where the fill occurred in the current trading session. Valid values: 4 = Normal trade 5 = Auction trade S = Combo-to-outright trade R = Combo-to-Combo trade A = Strip to strip B = Strip to outright.
54	Side	char	Y	Side of order. Valid values: 1 = Buy 2 = Sell
38	OrderQty	Qty	Y	Total order quantity.
44	Price	Price		Price per unit quantity entered on order.
31	LastPx	Price		Price of this (last) fill. Supplied if ExecType=Trade (150=F).
6	AvgPx	Price		Calculated average price of all fills on this order. Ignore for products quoted in Yield.
15	Currency	Currency		Identifies currency used for price.
32	LastQty	Qty		Quantity bought/sold on this (last) fill. Supplied if ExecType=Trade (150=F).

Tag	Name	Data Type	Reqd	Comment
14	CumQty	Qty	Y	Total quantity filled of all executions for the order.
151	LeavesQty	Qty	Y	Quantity open for further execution. Set to zero for cancelled orders.
75	TradeDate	LocalMktDate		Business trade date.
40	OrdType	char		Order type. Valid values: 2 = Limit.
59	TimeInForce	char		Specifies how long the order remains in effect. Absence of this field is interpreted as Day. Valid values: 0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK) 6 = Good Till Date (GTD).
29	LastCapacity	char		Used to indicate capacity of broker. Valid values for XSFE and NZFX: 3 = Cross as principal.
55	Symbol	String [255]		Common, human understood representation of the security.
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type.

Tag	Name	Data Type	Reqd	Comment
22	SecurityIDSource	String		Identifies the source of the SecurityID (48) value. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
207	SecurityExchange	Exchange		Market of security. Only returned if entered on the NewOrderSingle (D) or OrderCancelReplaceRequest (G). Valid values: XSFE = Sydney Futures Exchange NZFX = New Zealand Futures and Options Exchange.
18	ExecInst	MultipleCharValue		Instructions for handling the order. Valid values: o = Cancel on connection loss.
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> 448	PartyID	String [see comment]		Identifies the party. Required, if NoPartyIDs > 0. Required, if PartyIDSource is specified. The maximum number of characters depends on the PartyRole (452) value. See PartyRole (452).
> 447	PartyIDSource	char		Identifies class or source of the PartyID (448) value. Required, if PartyID is specified. Valid values: D = Proprietary/Custom code

Tag	Name	Data Type	Reqd	Comment
> 452	PartyRole	int		The role of the party in the transaction. Required if PartyID is specified. Maximum number of characters for each type in []. Valid values: 1 = Executing Firm [64] 4 = Clearing Firm [64] 7 = Entering Firm [64] 11 = Order Origination Trader [64] 12 = Executing Trader [64] 17 = Contra Firm [64] 24 = Customer Account [15] 29 = Intermediary [64] 36 = Entering Trader [64] 45 = Secondary Account Number [32] 76 = Desk ID (Identifies the shared order group that this order is part of). [10]
1057	AggressorIndicator	Boolean		Used to identify whether the order initiator is an aggressor or not in the trade. Valid values: Y = Order initiator is aggressor N = Order initiator is passive.
1624	NoMatchInst	NumInGroup		Number of instructions in the MatchingInstructions repeating group. MatchingInstructions are used to indicate whether UCP was applied to an order, or activated on a trade resulting from the order.

Tag	Name	Data Type	Reqd	Comment
> 1625	MatchInst	int		Matching instruction. Supplied if NoMatchInst > 0 Valid values: 1 = Match: Where ExecType=Trade (150=F), this trade* is handled as a private booking report; UCP keys on both sides of the trade matched. For other ExecType values, reflects the value entered on the order. 2 = Do Not Match: Only present where ExecType=Trade (150=F) and UCP keys did not match for this trade* * If a trade results from an order in a multi-leg product (e.g. a spread or UDC) only the individual legs of the product will trade and can be settled. The multi-leg product does not settle. The match instruction reported for such trades has no meaning. The ExecutionReport (8) received by the user(s) for the individual leg products indicates if UCP was applied or not.
> 1626	MatchAttribTagID	TagNum		This tag is required by the FIX protocol when NoMatchInst > 0. Valid values: 10000 = UCP Key.
> 1627	MatchAttribValue	String [10]		Value to apply the matching instruction. When MatchAttribTagID=10000, an integer number used for UCP - key supplied to activate Unintentional Crossing Protection (UCP).
432	ExpireDate	LocalMktDate		Date of order expiration (last day the order can trade), always expressed in terms of the local market date. Reported if TimeInForce is GTD and ExpireTime is not specified.

Tag	Name	Data Type	Reqd	Comment
126	ExpireTime	UTCTimestamp		Time/Date of order expiration (always expressed in UTC). Reported if TimeInForce is GTD and ExpireDate is not specified.
58	Text	String [see comment]		Free format text string. The maximum number of characters may exceed 128 for complex messages.
60	TransactTime	UTCTimestamp		Timestamp when the business transaction represented by the message occurred.
555	NoLegs	NumInGroup		Number of InstrumentLeg repeating group instances.
> 600	LegSymbol	String [255]		Symbol of the leg.
> 602	LegSecurityID	String [10]		SecurityID of the leg.
> 603	LegSecurityIDSource	String	C	Identifies class or source of the of the LegSecurityID value. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
> 1788	LegID	String [84]		Unique identifier assigned to this leg trade.
> 1366	LegAllocID	String [10]		Identifier assigned to a trade by ASX clearing purposes - clearing deal number.
> 637	LegLastPx	Price		Execution price assigned to a leg of a multileg instrument.
> 1418	LegLastQty	Qty		Fill quantity for the leg instrument. Formatted as per tag 38.
> 624	LegSide	char		The side of this individual leg.

Tag	Name	Data Type	Reqd	Comment
				Valid values: 1 = Buy 2 = Sell.
> 556	LegCurrency	Currency		Multileg instrument's currency.
> 685	LegOrderQty	Qty		Quantity ordered of this leg.
> 2254	NoLegInstrumentParties	NumInGroup		Number of parties in the repeating group. Parties involved in a leg trade. In anonymous markets, these tags are only populated when leg trade is a crossing, and contra firm will have the same value as executing firm.
> > 2255	LegInstrumentPartyID	String [64]		Identifies the party. Supplied if NoLegInstrumentParties > 0.
> > 2256	LegInstrumentPartyIDSource	char		Identifies class or source of the LegInstrumentPartyID (2255) value. Supplied if LegInstrumentPartyID is specified. Valid values: D = Proprietary/Custom code.
> > 2257	LegInstrumentPartyRole	int		The role of the party in the transaction. Required, if LegInstrumentPartyID is specified. Valid values: 1 = Executing Firm 17 = Contra Firm Additional values specified in PartyRole (452) may also be supplied.
	StandardTrailer		Y	

4.1.1. ExecutionReport (8) Examples

New Orders

8=FIXT.1.1|9=0000397|35=8|49=ASX|56=ABCD1|34=930|52=20161201-04:15:48.202|369=608|37=6209942839934074881|
198=6209942839934074881|11=ID-1480565748180|453=5|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|17=6209942839934074881-3724|150=0|39=0|1=ABC1|581=1|55=IBZ6|48=64757|
22=8|54=1|38=1|40=2|44=96.4|15=AUD|59=0|151=1|14=0|6=0|60=20161201-04:15:48.199|797=Y|10=252|

8=FIXT.1.1|9=0000397|35=8|49=ASX|56=ABCD1|34=931|52=20161201-04:15:52.199|369=608|37=6209942856702902273|
198=6209942856702902273|11=ID-1480565752181|453=5|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|17=6209942856702902273-3725|150=0|39=0|1=ABC1|581=1|55=IBZ6|48=64757|
22=8|54=1|38=2|40=2|44=96.4|15=AUD|59=0|151=2|14=0|6=0|60=20161201-04:15:52.197|797=Y|10=216|

Fills

8=FIXT.1.1|9=0000451|35=8|49=ASX|56=ABCD1|34=1042|52=20161201-04:43:12.212|369=719|37=6209942856702902273|
198=6209942856702902273|11=ID-1480565752181|453=5|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|880=1000000020|17=6209949735411793921-B|150=F|39=2|1=ABC1|581=1|
574=4|55=IBZ6|48=64757|22=8|54=1|38=2|40=2|44=96.4|15=AUD|59=0|1057=N|32=2|31=96.4|151=0|14=2|6=96.4|75=20161201|
60=20161201-04:43:12.209|797=Y|10=191|

8=FIXT.1.1|9=0000451|35=8|49=ASX|56=ABCD1|34=1045|52=20161201-04:43:44.933|369=721|37=6209943024261152769|
198=6209943024261152769|11=ID-1480565792128|453=5|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|880=1000000021|17=6209949872632643585-A|150=F|39=2|1=ABC1|581=1|
574=4|55=APH7|48=58950|22=8|54=2|38=2|40=2|44=5437|15=AUD|59=0|1057=N|32=2|31=5437|151=0|14=2|6=5437|75=20161201|
60=20161201-04:43:44.925|797=Y|10=208|

4.2. TradeCaptureReport (AE)

Trade Capture Report (AE) messages are used to transmit information about trades (fills).

The following should be noted:

- ClOrdID (11) is as entered on the original connection. For drop copy connections that report the activity of more than one FIX connection, these values may not be unique. OrderID (37), which is supplied by the exchange, is unique and can be used to identify the order.
- Parties block is used to communicate the entities involved and their roles within the message.
- For combination orders, each leg trade is reported in an individual message. StrategyLinkID (1851) can be used to identify all the trades for the match event.
- CopyMsgIndicator is included in all drop copy messages, and is set to Y.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = AE
571	TradeReportID	String [64]		Unique identifier of trade capture report.
487	TradeReportTransType	int	Y	Identifies Trade Report message transaction type. Valid values: 0 = New 1 = Cancel (used for trade busts)
568	TradeRequestID	String [64]		Unique Identifier for the trade request.

Tag	Name	Data Type	Reqd	Comment
880	TrdMatchID	String [10]		Identifier assigned to a trade for ASX clearing purposes - clearing deal number. For multi-leg product executions, this value is not sent to the clearing system. See individual legs for the clearing deal number for each leg.
55	Symbol	String [255]	Y	Common, human understood representation of the security.
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type.
22	SecurityIDSource	String		Identifies the source of the SecurityID (48) value. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
32	LastQty	Qty		Quantity bought/sold on this (last) fill.
31	LastPx	Qty		Price of this (last) fill.
75	TradeDate	LocalMktDate		Business trade date.
60	TransactTime	UTCTimestamp		Timestamp when the business transaction, represented by the message, occurred.
574	MatchType	String		The point in the matching process at which the trade was matched. Valid values: 4 = Normal trade 5 = Auction trade S = Combo-to-outright trade

Tag	Name	Data Type	Reqd	Comment
				R = Combo-to-Combo trade A = Strip to strip B = Strip to outright.
570	PreviouslyReported	Boolean		Indicates if the trade capture report was previously reported to the counterparty. Valid values: N = Not reported to counterparty
912	LastRptRequested	Boolean		Indicates whether this message is the last report message in response to a request message, e.g. TradeCaptureReportRequest (35=AD). Valid values: N = Not the last message Y = Last message
552	NoSides	NumInGroup		Number of Side repeating group instances.
> 54	Side	char		Side of order. Valid values: 1 = Buy 2 = Sell.

Tag	Name	Data Type	Reqd	Comment
> 29	LastCapacity	char		Used to indicate capacity of broker. Valid values for XSFE and NZFX: 3 = cross as principal.
> 1057	AggressorIndicator	Boolean		Used to identify whether the order initiator is an aggressor or not in the trade. Valid values: Y = Order initiator is aggressor N = Order initiator is passive.
> 1851	StrategyLinkID	String [19]		Identifies the multileg strategy (e.g. spread) to which the trade belongs. This links together trade legs executed as part of a strategy during a single match event.
> 453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> > 448	PartyID	String [see comment]		Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0. The maximum number of characters depends on the PartyRole (452) value. See PartyRole (452).
> > 447	PartyIDSource	char		Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Valid values: D = Proprietary/Custom code.

Tag	Name	Data Type	Reqd	Comment
> > 452	PartyRole	int		The role of the party in the transaction. Required if PartyID is specified. Maximum number of characters for each type in []. Valid values: 1 = Executing Firm [64] 4 = Clearing Firm [64] 7 = Entering Firm [64] 11 = Order Origination Trader [64] 12 = Executing Trader [64] 17 = Contra Firm [64] 24 = Customer Account [15] 29 = Intermediary [64] 36 = Entering Trader [64] 45 = Secondary Account Number [32] 76 = Desk ID (Identifies the shared order group that this order is part of). [10]
> 1624	NoMatchInst	NumInGroup		Number of instructions in the MatchingInstructions repeating group. For trades resulting from UCP orders, this block will be used to indicate if the trade resulted in private booking report (UCP activated) or a normal trade (UCP did not activate).
> > 1625	MatchInst	int		Matching instruction. Supplied if NoMatchInst > 0 Valid values: 1 = Match: This trade is handled as private booking report; UCP keys on both sides of the trade matched 2 = Do Not Match: UCP keys did not match for this trade.
> > 1626	MatchAttribTagID	TagNum		This tag is required by the FIX protocol when NoMatchInst > 0

Tag	Name	Data Type	Reqd	Comment
				Valid values: 10000 = UCP Key.
> > 1627	MatchAttribValue	String [10]		Value to apply the matching instruction. When MatchAttribTagID=10000, an integer number used for UCP - key supplied to activate Unintentional Crossing Protection (UCP).
> 1	Account	String [10]		Account supplied on the order.
> 581	AccountType	int		Type of account associated with the order. Valid values: 1 = Customer 2 = Non-customer.
> 37	OrderID	String [19]		Unique identifier for Order as assigned by ASX. Private to the member firm entering the order.
> 11	ClOrdID	String [128]		Identifier for Order as assigned by Client.
> 58	Text	String [see comment]		Free format text string. The maximum number of characters may exceed 128 for complex messages.
797	CopyMsgIndicator	Boolean		Indicates whether or not this message is a drop copy of another message. Always included in drop copy messages. Valid values: Y = Message is a drop copy.
	StandardTrailer		Y	

4.2.1. TradeCaptureReport (AE) Example

```
8=FIXT.1.1|9=0000441|35=AE|49=ASX|56=ABCD1|34=1051|52=20161201-04:44:53.103|369=727|571=6209608690589253633-B|
487=0|568=ABC-AD-20161201-04:44:50|912=N|880=1000000001|570=N|55=IRZ6|48=65014|22=8|32=1|31=96.4|75=20161201|
60=20161130-06:08:00.787|574=5|552=1|54=1|453=7|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ABC|447=D|452=17|448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|1=ABC1|581=1|1057=N|37=6209606224527245313|
11=ABCO1-80|29=3|797=Y|10=157|

8=FIXT.1.1|9=0000441|35=AE|49=ASX|56=ABCD1|34=1052|52=20161201-04:44:53.104|369=727|571=6209608690589253633-A|
487=0|568=ABC-AD-20161201-04:44:50|912=N|880=1000000001|570=N|55=IRZ6|48=65014|22=8|32=1|31=96.4|75=20161201|
60=20161130-06:08:00.787|574=5|552=1|54=2|453=7|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ABC|447=D|452=17|448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|1=ABC1|581=1|1057=N|37=6209606224988618753|
11=ABCO1-89|29=3|797=Y|10=191|

8=FIXT.1.1|9=0000441|35=AE|49=ASX|56=ABCD1|34=1053|52=20161201-04:44:53.104|369=727|571=6209608690589253634-B|
487=0|568=ABC-AD-20161201-04:44:50|912=N|880=1000000002|570=N|55=IRZ6|48=65014|22=8|32=1|31=96.4|75=20161201|
60=20161130-06:08:00.787|574=5|552=1|54=1|453=7|448=ABC|447=D|452=1|448=ABCO1|447=D|452=12|448=ABC|447=D|452=7|
448=ABC|447=D|452=17|448=ASXCL|447=D|452=4|448=ABCO1|447=D|452=36|1=ABC1|581=1|1057=N|37=6209606224527245313|
11=ABCO1-80|29=3|797=Y|10=162|
```

4.3. PartyRiskLimitsReport (CM)

There is a preconfigured warning threshold for accounts related to Pre-Trade Risk Management (PTRM) limits.

The Party Risk Limits Report message is used to inform subscribers that the threshold has been exceeded for their PTRM limits. This can be sent at an account or firm level. Access to receive this message may be limited by the permissions set for the subscriber.

A PartyRiskLimitsReport (CM) message is sent for every order that breaches a PTRM threshold. If an order simultaneously breaches the lower and upper threshold, only one message is sent. Similarly, if a spread order breaches both the long and short limits, only one message, notifying of the long breach, will be sent. The message will contain both the actual utilisation and the upper limit threshold value. If an order breaches the PTRM limit, the order will be rejected, and a PartyRiskLimitsReport (CM) message will not be sent.

The lower limit is set at 75%, the upper one at 90%.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = CM
1128	ApplVerID	String [128]		Identifies the Service pack release Valid values: 9 = FIX50SP2
1667	RiskLimitReportID	String [128]	Y	Identifier for the PartyRiskLimitsReport.
1677	NoPartyRisksLimits	NumInGroup		Number of party risk limits.
> 1671	NoPartyDetails	NumInGroup		Number of parties that this limit applies to. Required if NoPartyRiskLimits > 0.

Tag	Name	Data Type	Reqd	Comment
> > 1691	PartyDetailID	String [see comment]		Identifies the party. Required if NoPartyDetails > 0. The maximum number of characters depends on the PartyDetailRole (1692) value. See PartyDetailRole (1692).
> > 1692	PartyDetailIDSource	char		Identifies the source of the PartyDetailID value. Required when NoPartyRiskLimits > 0. Valid values: D = Proprietary/Custom code.
> > 1693	PartyDetailRole	int		Identifies the role of the party. Required when NoPartyRiskLimits > 0. Valid values: 1 = Executing firm. Up to 64 characters. 24 = Account (Account (1) value entered on the order when the risk applies to the account. Up to 10 characters.)
> 1669	NoRiskLimits	NumInGroup		Number of risk limits.
> > 1529	NoRiskLimitTypes	NumInGroup		Number of risk limits and associated warnings.
> > > 1530	RiskLimitType	int		Used to specify the type of risk limit. Required, if NoRiskLimitTypes > 0. Valid values: 4 = Long limit 5 = Short limit.
> > > 1765	RiskLimitUtilizationPercent	Percentage		Percentage of utilisation of a party's set risk limit.

Tag	Name	Data Type	Reqd	Comment
> > > 1767	RiskLimitAction	int		Risk protection action. Valid values: 4 = Warning
> > > 1559	NoRiskWarningLevel	NumInGroup		Number of risk warning levels.
> > > > 1769	RiskWarningLevelAction	int		Action to be taken when warning level is breached. Valid values: 4 = Warning.
> > > > 1560	RiskWarningLevelPercent	Percentage		Percent of risk limit at which a warning is issued.
> > > 1534	NoRiskInstrumentScopes	NumInGroup		Number of instruments the risk limit applies to.
> > > > 1535	InstrumentScopeOperator	int		Specifies the instruments included in the limit. Valid values: 1 = Include.
> > > > 1545	InstrumentScopeSecurityGroup	String [128]		Commodity code of securities that the risk limit applied to, e.g. XT.
> > > > 1616	InstrumentScopeSecurityExchange	Exchange		SecurityExchange the limit applies to. Valid values: XSFE = Sydney Futures Exchange NZFX = New Zealand Futures and Options Exchange.
60	TransactTime	UTCTimestamp		Timestamp when the business transaction, represented by the message, occurred.

Tag	Name	Data Type	Reqd	Comment
58	Text	String [see comment]		Text string describing warning. The maximum number of characters may exceed 128 for complex error messages.
	StandardTrailer		Y	As defined in standard trailer.

4.3.1. PartyRiskLimitReport (CM) Example

```
8=FIXT.1.1|9=0000301|35=CM|1128=9|49=ASX|56=ABCD1|34=1401|52=20161201-05:18:25.663|369=864|
1667=1262606_256_ABCD1_7|1677=1|1671=1|1691=ABC1|1692=D|1693=24|1669=1|1529=1|1530=4|1767=4|1765=0.79|1559=1|
1769=4|1560=0.75|1534=1|1535=1|1545=AP-S_SYMB|60=20161201-05:18:25.660|58=ABC1 exceeds 75.00%. Actual utilization
78.60%.|10=012|
```

4.4. TradeCaptureReportRequest (AD)

Drop copy connections provide the ability to request trades from previous days. This is done with the TradeCaptureReportRequest (AD) message. Trades are returned in TradeCaptureReport (AE) messages. If the request cannot be satisfied, a TradeCaptureReportRequestAck (AQ) message will be returned indicating the reason the request could not be honoured.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = AD
568	TradeRequestID	String [64]	Y	Unique identifier for the trade capture report request.
569	TradeRequestType	int	Y	Type of trade capture report requested. Valid values: 0 = All Trades.
55	Symbol	String [255]		Common, human understood representation of the security. Omit Symbol (55) to request trades in all instruments. Set to [N/A] if specifying instrument using SecurityID (48).
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22). Request will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A].
22	SecurityIDSource	String		Identifies the source of the SecurityID (48) value. Required if SecurityID (48) is specified. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.

Tag	Name	Data Type	Reqd	Comment
580	NoDates	NumInGroup		Number of dates to report on. Up to two groups can be specified. First group controls when trades are to be reported from. Second group controls when trades are to be reported to.
> 75	TradeDate	LocalMktDate	C	Business Trade Date of trade report. In the first group: First Business Trade Date to report trades from. In the second group: Last Business Trade Date to report trades until. Required when NoDates > 0. Value is ignored if TransactTime (60) is supplied.
> 60	TransactTime	UTCTimestamp		Time trade occurred. In the first group: Earliest time of trade to be returned. In the second group: Last time of trade to be returned. If TransactTime (60) is supplied, TradeDate (75) will be ignored. TradeDate (75) must be present to define the repeating group instance.
	StandardTrailer		Y	

4.4.1. TradeCaptureReportRequest (AD) Example

```
8=FIXT.1.1|9=103|35=AD|49=ABCD1|56=ASX|34=727|52=20161201-04:44:50|568=ABC-AD-20161201-04:44:50|569=0|580=1|
75=20161201|10=003|
```

4.5. TradeCaptureReportRequestAck (AQ)

In the Trade Capture Report, the ack will be sent in the following circumstances:

- To indicate a successful request for which there are no trades to return
- If a request cannot be satisfied, to notify the user why it cannot be met.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = AQ
568	TradeRequestID	String [64]	Y	Identifier of the trade capture report request.
569	TradeRequestType	int	Y	Type of trade capture report requested. Valid values: 0 = All Trades.
748	TotNumTradeReports	int		Total number of trade reports returned.
749	TradeRequestResult	int	Y	Result of Trade Request Valid values: 1 = Invalid or Unknown Instrument 0 = Successful 8 = Type not supported 99 = Other.

Tag	Name	Data Type	Reqd	Comment
750	TradeRequestStatus	int	Y	Status of Trade Request. Valid values: 1 = Completed 2 = Rejected.
55	Symbol	String [255]		Common, human understood representation of the security.
58	Text	String [see comment]		Text describing the reason for rejection. The maximum number of characters may exceed 128 for complex error messages.
	StandardTrailer		Y	

4.5.1. TradeCaptureReportRequestAck (AQ) Examples

No records returned

```
8=FIXT.1.1|9=0000125|35=AQ|49=ASX|56=ABCD1|34=1189|52=20161201-04:46:12.499|369=734|568=ABC-AD-20161201-04:46:12|
569=0|748=0|749=0|750=1|55=[N/A]|10=002|
```

Invalid symbol on request

```
8=FIXT.1.1|9=112|35=AD|49=ABCD1|56=ASX|34=797|52=20161201-05:01:39|568=ABC-AD-20161201-05:01:39|569=0|580=1|
75=20161201|55=IRZ67|10=022|

8=FIXT.1.1|9=0000119|35=AQ|49=ASX|56=ABCD1|34=1252|52=20161201-05:01:39.485|369=797|568=ABC-AD-20161201-05:01:39|
569=0|749=1|750=2|55=[N/A]|10=243|
```

4.6. QuoteRequest (R)

QuoteRequest (R) messages provide information on accepted RFQs. Details of the user who entered the QuoteRequest (R) are in the RootParties block. QuoteReqID (131) is the identifier entered by the user and therefore duplicates may occur if different users use the same identifiers.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = R
131	QuoteReqID	String [64]	Y	Identifier of QuoteRequest (R) entered by the FIX user. [N/A] if quote was created on the ASX Trading Terminal.
1116	NoRootPartyIDs	NumInGroup		Number of parties that this quote request applies to.
> 1117	RootPartyID	String [see comment]		Identifies the party Required if NoRootPartyIDs > 0 The maximum number of characters depends on the RootPartyRole (1119) value. See RootPartyRole (1119).
> 1118	RootPartyIDSource	char		Identifies the source of the RootPartyID value. Required when NoRootPartyIDs > 0 Valid values: D = Proprietary/Custom code
> 1119	RootPartyRole	int		Identifies the role of the party. Maximum number of characters for each type in []. Required when NoRootPartyIDs > 0 Valid values: 7 = Entering firm [64]

Tag	Name	Data Type	Reqd	Comment
				36 = Entering trader [64]
146	NoRelatedSym	NumInGroup	Y	Number of related symbols (instruments) in this request.
> 55	Symbol	String [255]		Common, human understood representation of the security.
> 48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type.
> 22	SecurityIDSource	String		Identifies the source of the SecurityID (48) value. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
> 54	Side	char		Side of order. If Side is omitted, indicates a two sided quote. Valid values: 1 = Buy 2 = Sell 8 = Crossing.
> 38	OrderQty	Qty		Total quantity
> 60	TransactTime	UTCTimestamp		Timestamp when the quote entered by the user was accepted by the matching engine.
	StandardTrailer		Y	

4.6.1. QuoteRequest (R) Example

```
8=FIXT.1.1|9=0000193|35=R|49=ASX|56=ABCD1|34=211|52=20170605-02:17:16.327|369=208|131=QR-174|1116=2|1117=ABC|
1118=D|1119=7|1117=ABCO1|1118=D|1119=36|146=1|55=ARH8|48=92719|22=8|54=1|38=326|60=20170605-02:17:16.325|10=101|
```

5. General Messages

The following sections cover the supported general messages.

5.1. News (B)

The News (B) message is used to disseminate text information to subscribers.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = B
148	Headline	String [see comment]	Y	Description of information being transmitted. The maximum number of characters may exceed 128 for complex messages.
42	Orig Time	UTCTimestamp	Y	Time of message origination.
33	NoLinesOfText	NumInGroup	Y	Number of Test (58) repeating group instances.
> 58	Text	String [see comment]	Y	Detailed description, or repetition of Headline (148) if no additional information to be transmitted. The maximum number of characters may exceed 128 for complex messages.
	StandardTrailer		Y	

5.2. BusinessMessageReject (j)

The BusinessMessageReject message can reject an application-level message, which fulfills session-level rules and cannot be rejected via any other means; typically unsupported application messages or application messages lacking a specific reject message. If the message fails, a session-level rule (for example, body length is incorrect), a session-level Reject message should be issued.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = j (lowercase)
45	RefSeqNum	SeqNum		MsgSeqNum of rejected message.
372	RefMsgType	String	Y	The MsgType of the FIX message being referenced.
380	BusinessRejectReason	int	Y	Code to identify reason for a Business Message Reject message. Valid values: 3 = Unsupported Message Type 4 = Application not available
58	Text	String [see comment]		Where possible, message to explain reason for rejection. The maximum number of characters may exceed 128 for complex error messages.
	StandardTrailer		Y	

5.2.1. BusinessMessageReject (j) Example

```
8=FIXT.1.1|9=0000109|35=j|49=ASX|56=ABCD1|34=1263|52=20161201-05:04:20.089|369=807|45=808|372=D|380=3|
58=Unsupported Message Type|10=244|
```

Disclaimer

This document provides general information only and may be subject to change at any time without notice. ASX Limited (ABN 98 008 624 691) and its related bodies corporate ("ASX") makes no representation or warranty with respect to the accuracy, reliability or completeness of this information. To the extent permitted by law, ASX and its employees, officers and contractors shall not be liable for any loss or damage arising in any way, including by way of negligence, from or in connection with any information provided or omitted, or from anyone acting or refraining to act in reliance on this information. The information in this document is not a substitute for any relevant operating rules, and in the event of any inconsistency between this document and the operating rules, the operating rules prevail to the extent of the inconsistency.

ASX Trade Marks

The trademarks listed below are trademarks of ASX. Where a mark is indicated as registered it is registered in Australia and may also be registered in other countries. Nothing contained in this document should be construed as being any licence or right to use of any trade mark contained within the document.