

ASX 24 FIX Order Entry Specification

Version 2.05 | October 2023



Table of Contents

1.	FIX Order Entry Introduction	5
1.1.	Document Information	5
1.1.1.	Terms and Acronyms	5
1.1.2.	Usage of String Datatypes	6
1.1.3.	Usage of UTCTimestamp	6
1.1.4.	Handling of unsupported messages and tags	6
1.1.5.	Definition of required column values	6
1.2.	Version History	7
2.	Common Message Structures	9
2.1.	Standard Header	9
2.2.	Standard Trailer	11
3.	FIX Session Layer	12
3.1.	Overview	12
3.2.	Connecting to the FIX Server Process	12
3.3.	Lost connection	13
3.4.	Logon (A)	14
3.5.	Heartbeat (0)	18
3.6.	TestRequest (1)	19
3.7.	ResendRequest (2)	20
3.8.	Reject (3)	21
3.9.	SequenceReset (4)	23
3.10.	Logout (5)	24
3.11.	BusinessMessageReject (j)	26
4.	Application Messages	27
4.1.	Control of orders for Shared Order Groups	27
4.1.1.	Summary of Identification of Orders in FIX	27
4.1.2.	Default user and Shared Order Groups	27
4.1.3.	Control of orders across multiple connections	28
4.1.4.	Example 1	28
4.1.5.	Example 2	28
4.1.6.	Example 3	29
4.1.7.	Cancel Orders on Connection Loss	29



4.1.8.	Disabling a SOG Owner User	30
4.2.	Interpreting PartyRole Values	30
4.2.1.	PartyRole (452) Usage	30
4.2.2.	Examples.....	31
4.3.	Cancel on Disconnect.....	33
4.3.1.	Identifying disconnection	34
4.4.	NewOrderSingle (D)	35
4.4.1.	NewOrderSingle (D) Example – SOG and Cancel on Disconnect.....	40
4.4.2.	NewOrderSingle (D) Example – UCP Key Usage	41
4.4.3.	NewOrderSingle (D) Example – GTD Order, Specifying Date	41
4.5.	OrderCancelRequest (F).....	42
4.5.1.	OrderCancelRequest (F) Example – Basic Cancel.....	45
4.5.2.	OrderCancelRequest (F) Example – SOG Cancel.....	45
4.6.	OrderCancelReplaceRequest (G).....	47
4.6.1.	OrderCancelReplaceRequest (G) Example – Amend of TIF and Quantity.....	54
4.6.2.	OrderCancelReplaceRequest (G) Example – SOG Amend.....	54
4.7.	ExecutionReport (8)	55
4.7.1.	ExecutionReport (8) Example – Futures Contract, Market to Limit with UCP Key Applied	68
4.7.2.	ExecutionReport (8) Example – Spread Contract.....	69
4.7.3.	ExecutionReport (8) Example – Order Expires Due to TIF Settings.....	70
4.7.4.	ExecutionReport (8) Example – Restatement.....	70
4.7.5.	Expiry of Orders.....	70
4.7.6.	Restatement of Long Orders	72
4.8.	OrderCancelReject (9).....	73
4.8.1.	OrderCancelReject (9) Example	75
4.9.	QuoteRequest (R)	76
4.9.1.	QuoteRequest (R) Example.....	77
4.10.	QuoteRequestReject (AG).....	78
4.10.1.	QuoteRequestReject (AG) Example	79
4.11.	SecurityDefinitionRequest (c)	80
4.11.1.	SecurityDefinitionRequest (c) Example	82
4.12.	SecurityDefinition (d)	83
4.12.1.	Sorting of legs in UDCs.....	83
4.12.2.	SecurityDefinition (d) Example – Accept As Is	84



4.12.3. SecurityDefinition (d) Example – Accept As with Revisions	85
4.12.4. SecurityDefinition (d) Example – Reject	85
4.13. OrderMassStatusRequest (AF)	86
4.13.1. OrderMassStatusRequest (AF) Example – No SOG Specified	88
4.13.2. OrderMassStatusRequest (AF) Example – SOG Specified	89
4.13.3. OrderMassStatusRequest (AF) Example – No Results	90
4.14. OrderMassCancelRequest (q)	91
4.14.1. OrderMassCancelRequest (q) – No SOG	93
4.14.2. OrderMassCancelRequest (q) with SOG	93
4.15. OrderMassCancelReport (r)	95
4.15.1. OrderMassCancelReport (r)	97
4.16. PartyRiskLimitsReport (CM)	98
4.16.1. PartyRiskLimitsReport (CM) Example	101
5. General Messages	102
5.1. News (B)	102
5.1.1. News (B) Example	102
6. Message Sequencing	103
6.1. Multiple gateways	103
6.2. Multiple sessions per gateway	103
6.3. Multiple instruments per session	104
6.4. Matching engine	104

1. FIX Order Entry Introduction

ASX provides the facility to create and manage orders using the FIX protocol. This document provides guidance on using FIX protocol messages to:

- Establish and maintain a FIX session
- Create, amend, and cancel orders
- Receive information about orders and trades
- Create user defined combination instruments.

These rules of engagement are based on the FIX 5.0 SP2 Specification, Extension Pack 196 (FIX 5.0 SP2 EP 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 the scope of this document, and therefore customers should refer to the [FIX Trading Community](#) for a full understanding of the protocol, prior to using this guide.

Information on other aspects of operating the service such as setting up pre-trade risk management, trading protection limits, and failover can be found on the ASX website.

1.1. 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.

1.1.1. 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

1.1.2. 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 comment* 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.

1.1.3. 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.

1.1.4. 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.

1.1.5. 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.

1.2. 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.
V1.03	December 2016	Added examples to Application Messages, 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: <ul style="list-style-type: none"> • ExecutionReport (8) • NewOrderSingle (D) • OrderCancelReplaceRequest (G) • OrderMassStatusRequest (AF)
V1.06	May 2017	<ul style="list-style-type: none"> • Clarification on use of the Party Role of Entering Trade when using Shared Order Groups (SOGs): <ul style="list-style-type: none"> – New section 4.1.2 Default user and Shared Order Groups – Clarification to OrderMassStatusRequest (AF) and OrderMassCancelRequest (q) – OrderCancelReject (9) changes for maintenance release
V2.00	July 2017	Changes for the Service Release <ul style="list-style-type: none"> • New CxlRejReason (102) value for OrderCancelReject (9) • New OrdRejReason (103) value for ExecutionReport (8) • Correction of order expiry when ExpireTime (126) is set • Removal <i>PartyRole of Contra clearing firm</i> (452=18)
V2.01	August 2017	<ul style="list-style-type: none"> • Clarify meaning of MatchInst (1625) on multileg trades • Indicate that UCP key must be reapplied in OrderCancelReplaceRequest (G) messages • Detail how unsupported messages and tags are handled
V2.02	December 2017	<ul style="list-style-type: none"> • Clarify the contents of MatchAttribValue (1627) • Removal CxlRejReason value 18 • Added information about message sequencing



Version	Date	Comment
V2.03	August 2018	<ul style="list-style-type: none"> Clarify what UTCTimestamp format ASX 24 accepts Additional information on the removal of cancel on disconnect orders
V2.04	April 2020	<ul style="list-style-type: none"> Section 4.4, Clarification of tag 126 functionality
V2.05	October 2023	<ul style="list-style-type: none"> Amended references of 'NTP' to 'ASX 24'



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	String [64]	Y	Identifies the sender of the message.
56	TargetCompID	String [64]	Y	Identifies the receiver of the message.
115	OnBehalfOfCompID	String [64]		Value used to identify user subject to OrderMassCancelRequest.
34	MsgSeqNum	SeqNum	Y	Message sequence number.



Tag	Name	Data Type	Reqd	Comment
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.
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 a 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). 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. Overview

Exchange of FIX messages is organized into sessions and connections. This section describes how these are defined and used by the ASX. Please read this in conjunction with the FIX Session Protocol document.

On ASX 24, FIX sessions start on Sunday night when the trading system starts up. This enables one FIX session to encompass an entire ASX 24 trading week, ending on Saturday morning. Incoming and outgoing sequence numbers are set to 1, and messages sent prior to the start of the session are no longer available to be resent.

Within a FIX session, a client may choose to have one or more FIX connections, that is, logon, exchange messages, and logout (or other disconnection). Any identifiers that are defined as unique within a FIX session, such as ClOrdID (11), may not be reused within a FIX session. Either side may request the other to resend any message, at any point in the session.

A client may specify `ResetSeqNumFlag=true (141=Y)` on login. This will cause a new FIX session to be started and previous messages will no longer be available to be resent. The identifiers defined as unique within a FIX session, such as ClOrdID (11), will be unique across all the FIX sessions in an ASX 24 trading week. For example, a ClOrdID (11) for a day order used in an earlier FIX connection may not be reused in a subsequent connection established by setting `ResetSeqNumFlag=true (141=Y)`.

Note that setting the `ResetSeqNumFlag=true (141=Y)` after the first session state change may result in message loss. Specifically, any messages queued for sending to the client, such as, restatements, fills of long orders, and so on, will no longer be available for resending.

3.2. 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.



3.3. 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) is 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.



3.4. 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 reattempt logon 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, and so on.

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

Recovery is supported using the ResendRequest (2) and using the 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 the FIX server detects but is unable to 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	<p>Method of encryption.</p> <p>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.</p> <p>Valid values: 0 = None/Other.</p>
108	HeartBtInt	int	Y	<p>Heartbeat interval in seconds. The same value is used by both sides.</p> <p>Specified by the logon initiator and reflected by the acceptor (ASX).</p> <p>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.</p>



Tag	Name	Data Type	Reqd	Comment
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.
789	NextExpectedMsgSeqNum	SeqNum		Next expected MsgSeqNum value to be received. Required when connecting to the secondary (standby) FIX Order Entry 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: <ul style="list-style-type: none"> • Contain an English upper case character (A-Z) • Contain an English lower case character (a-z) • Contain a Hindu Arabic numeral (0-9)



Tag	Name	Data Type	Reqd	Comment
				<ul style="list-style-type: none"> 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. <p>Passwords must be a minimum of 8 characters in length. 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.</p>
1409	SessionStatus	Int		<p>Status of the FIX session. Sent by ASX. Ignored if input by client.</p> <p>Valid values: 0 = Session active 1 = Session password changed.</p>
1137	DefaultAppVerID	String	Y	<p>Specifies the service pack release being applied to the message at the session level.</p> <p>Valid values: 9 = FIX50SP2</p>
58	Text	String [see comment]		<p>Free format text string</p> <p>Up to 128 characters on incoming message. Outgoing messages may exceed 128 characters.</p>
	StandardTrailer		Y	



3.5. 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	



3.6. 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	

3.7. 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	



3.8. 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. The 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	



3.9. 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	



3.10. 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 instance, 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). 103 = MsgSeqNum (34) must be 1 when ResetSeqNumFlag=true (141=Y) 104 = HeartBtInt (108) must be greater than 1



Tag	Name	Data Type	Reqd	Comment
				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	



3.11. BusinessMessageReject (j)

The BusinessMessageReject message can reject an application-level message which fulfils 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 (e.g. 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: 0 = Other - see Text (58) for further details 3 = Unsupported Message Type.
379	BusinessRejectRefID	String		The value of the business-level "ID" field on the message being referenced. Required unless the corresponding ID field was not specified. The list of id fields is in the description of the BusinessMessageReject (j) message in Volume 1 of FIX 5.0 SP2 specification.
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	



4. Application Messages

4.1. Control of orders for Shared Order Groups

ASX offers Shared Order Group (SOG) functionality to enable selected users to control orders entered by group members. Use of this functionality has the following implications for the identification and cancellation of orders.

4.1.1. Summary of Identification of Orders in FIX

A client uses the ClOrdID (11) to identify each instruction it sends, for example, new order, amend, cancel, and so on. The recipient uses OrderID (37) to identify the order once it has accepted it. The ClOrdID (11) is different from OrderID (37), in that it applies to one instruction, whereas OrderID (37) remains for the life of the order. When modifying an order, an instruction will contain a new ClOrdID (11) value and refer to the last instruction to touch the order (new or amend) in OrigClOrdID (41).

ClOrdID (11) is unique within a FIX session. If a client has multiple connections to the same counterparty, they may use the same values.

4.1.2. Default user and Shared Order Groups

The Party Role of Entering Trader (452=36) indicates the user that last acted on the order. When the OrderMassStatusRequest (AF) and OrderMassCancelRequest (q) requests are issued they will affect orders where the Entering Trader of the order matches the user id of the user issuing the request, irrespective of whether a SOG is specified.

For example: users A and B belong to the same SOG. User A places an order in this SOG and the Entering Trader is stamped as A. Both users issue an OrderMassStatusRequest (AF) without specifying the SOG. The order will be returned for A but not for B.

User B then modifies the order and the Entering Trader changes to B. Both users issue the same OrderMassStatusRequest (AF). This time the order is returned for user B but not for A.



4.1.3. Control of orders across multiple connections

When controlling a SOG order, a key requirement is to be able to identify the order. FIX sessions that did not create the order cannot use ClOrdID (11) as each connection has its own set of ClOrdID (11) values and does not know about other connections. OrderID (37) is unique across all orders and so can be used to uniquely identify an order.

To modify an order, FIX connections can:

- Specify OrderID (37), omitting OrigClOrdID (41)
- Specify OrigClOrdID (41), omitting OrderID (37). The value of OrigClOrdID (41) must be the value of ClOrdID (11) used by the last touch of the order by this FIX connection.
- Specify OrderID (37) and OrigClOrdID (41). OrigClOrdID (41) will be ignored and OrderID (37) used to identify the order.

4.1.4. Example 1

User A enters, User B amends using OrderID (37)

Action	FIX User A	FIX User B	Comment
Enter Order	11=A.1		
Execution Report	11=A.1 37=o.1	11=A.1 37=o.1	
Amend Order		11=B.1 37=o.1	
Execution Report	11=B.1 37=o.1	11=B.1 37=o.1	

4.1.5. Example 2

User A enters, User B attempts amend using user A ClOrdID (11).



Action	FIX User A	FIX User B	Comment
Enter Order	11=A.1		
Execution Report	11=A.1 37 = o.1	11=A.1 37=o.1	
Amend Order		11=B.1 41=A.1	OrderID (37) not specified.
Reject	<nothing sent to A>	Reject - unknown order	

4.1.6. Example 3

User A enters, User B amends, User A attempts to use previous ClOrdID (11) to modify order.

Action	FIX User A	FIX User B	Comment
Enter Order	11=A.1		
Execution Report	11=A.1 37=o.1	11=A.1 37=o.1	
Amend Order		11=B.1 37=o.1	
Execution Report	11=B.1 37=o.1	11=B.1 37=o.1	
Amend order	11=A.2 41=A.1		Uses the last ClOrdID that User A knows about.
Execution Report	11=A.2 41=A.1 37=o.1	11=A.2 37=o.1	

4.1.7. Cancel Orders on Connection Loss

When a user who had the last touch on an order (containing a cancel on disconnect parameter) disconnects, their order will be cancelled. If any other member of the SOG disconnects, the order remains on the market.



4.1.8. Disabling a SOG Owner User

When a user is disabled using the Admin and Risk Terminal (ART), all orders previously placed by that user are cancelled. Additionally, if this user is the owner user for one or more SOGs, all orders attached to those SOGs are cancelled.

4.2. Interpreting PartyRole Values

The Parties block (including TargetParties, LegInstrumentParties) is used to describe the entities and actors involved in a transaction. There are a large number of potential entities defined by FIX. This document describes the entities used by the ASX.

4.2.1. PartyRole (452) Usage

PartyRole values used by the ASX are described in the table below.

Note: The term member/member unit in the following descriptions refers to the member structure displayed in the ASX Risk Terminal.

Value	FIX User A	FIX User B	Example
1	Executing Firm	The member or member unit that executes the order or trade. If the order is marked as a Shared Order Group order, the member or member unit of the user that owns the shared order group.	ABC
4	Clearing Firm	The clearing member for this transaction.	ASXCL
7	Entering Firm	The member that places the order. If a member unit placed the order, the member that the member unit belongs to.	ABC
11	Order Origination Trader	Not currently used by ASX, or reflected back to the client.	
12	Executing Trader	The user within the member that owns order. For example, the name of the FIX connection or ASX Terminal user. If the order is marked as a SOG order, this will be the user that owns the shared order group.	ABC01



Value	FIX User A	FIX User B	Example
17	Contra Firm	The member that is on the other side of the transaction.	
24	Customer Account	Supplied by the client, reflected by ASX. Up to 15 characters. First six characters passed to the clearing system for STP usage.	
29	Intermediary	Not currently used by ASX, or reflected back to the client.	
36	Entering Trader	The entity within the member that placed the order or last acted on it.	ABCO2
45	Secondary Account Number	Supplied by the client, reflected by ASX. Can be used as a secondary client order identifier.	
76	Desk ID	Supplied by the client, reflected by ASX. Number that identifies the Shared Order Group that the order is part of.	2179

4.2.2. Examples

PartyRole (452) values are useful in determining who performed an action. This is particularly important when Shared Order Groups or Drop Copy connections are involved. The following examples show typical values that appear on messages from the ASX, reporting on orders or trades. Please note that the following setup was used to generate the examples.

Item	Value
Shared order group id	2176 Owner: ABCO6 Members: ABCT6, ABCMU1O1, ABCMU1T1
Shared order group id	2177 Owner: ABCMU1O1 Members: ABCT6, ABCO6, ABCMU1T1
Firm	ABC



Item	Value
Member unit	ABCMU1
Users (Firm/Member Unit)	ABCO6 (ABC), ABCT6 (ABC), ABCMU1O1 (ABCMU1), ABCMU1T1 (ABCMU1)
Clearer	ASXCL

The table below lists scenarios that produce responses from the ASX containing corresponding PartyRole values:

Scenario	Executing Firm (1)	Entering Firm (7)	Executing Trader (12)	Entering Trader (36)	Sample Party Block
Member unit user (ABCMU1O1) enters order	ABCMU1	ABC	ABCMU1O1	ABCMU1O1	453=5 448=ABCMU1 447=D 452=1 448=ABCMU1O1 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=ABCMU1O1 447=D 452=36
Member unit user (ABCMU1O1) enters SOG (2176) order, where does not own SOG (Member)	ABC	ABC	ABCO6	ABCMU1O1	453=6 448=ABC 447=D 452=1 448=ABCO6 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2176 447=D 452=76 448=ABCMU1O1 447=D 452=36
Member unit user (ABCMU1O1) enters SOG (2177) order, where owns SOG (Member unit)	ABCMU1	ABC	ABCMU1O1	ABCMU1O1	453=6 448=ABCMU1 447=D 452=1 448=ABCMU1O1 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2177 447=D 452=76 448=ABCMU1O1 447=D 452=36



Scenario	Executing Firm (1)	Entering Firm (7)	Executing Trader (12)	Entering Trader (36)	Sample Party Block
Member unit user (ABCMU1T1) enters SOG (2177) order, where does not own SOG (Member unit)	ABCMU1	ABC	ABCMU1O1	ABCMU1T1	453=6 448=ABCMU1 447=D 452=1 448=ABCMU1O1 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2177 447=D 452=76 448=ABCMU1T1 447=D 452=36
Member user (ABCO6) enters order	ABC	ABC	ABCO6	ABCO6	453=5 448=ABC 447=D 452=1 448=ABCO6 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=ABCO6 447=D 452=36
Member user (ABCO6) enters SOG (2176) order, where owns SOG	ABC	ABC	ABCO6	ABCO6	453=6 448=ABC 447=D 452=1 448=ABCO6 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2176 447=D 452=76 448=ABCO6 447=D 452=36
Member user (ABCO6) enters SOG (2177) order, where does not own SOG (Member unit)	ABCMU1	ABC	ABCMU1O1	ABCO6	453=6 448=ABCMU1 447=D 452=1 448=ABCMU1O1 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2177 447=D 452=76 448=ABCO6 447=D 452=36
Member user enters (ABCT6) SOG (2176) order, where does not own SOG (Member)	ABC	ABC	ABCO6	ABCT6	453=7 448=ABC 447=D 452=1 448=ABCO6 447=D 452=12 448=ABC 447=D 452=7 448=ASXCL 447=D 452=4 448=2176 447=D 452=76 448=ACME1 447=D 452=24 448=ABCT6 447=D 452=36

4.3. Cancel on Disconnect

When a FIX Order Entry connection is closed, any active orders marked as “Cancel on Disconnect” will be cancelled if the order book is in a state that allows order cancellation. If the order book is not in a state that allows order cancellation, the orders will be cancelled when the order book moves to a



state that allows cancellations, if the user has not re-established a connection. The order will also be cancelled if the system attempts to restore the order and the user has not re-established a connection.

4.3.1. Identifying disconnection

Cancellation processing is triggered as soon as the server detects the loss of the connection.

The reason for the FIX connection being closed can include several scenarios: explicit logout by either client or exchange, network interruption, and certain software and hardware component failures.

-  **Note:** In the event of loss of the FIX server the connection is considered lost after a configurable period (currently 75s). During this period, the customer may re-establish a session on the secondary FIX server, which will disable the cancellation of orders i.e. orders will remain in the market. In the event of any failure and recovery, it is always recommended that applications perform an order status request to confirm all active orders.

4.4. NewOrderSingle (D)

The New Order Single (D) message is used to submit an order to the exchange for execution. If accepted or rejected by the exchange, an Execution Report (8) message will be returned.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = D
11	ClOrdID	String [128]	Y	Unique identifier for Order as assigned by the client. Uniqueness must be guaranteed within a single trading session by the sender for clients who do not submit long dated orders (GTC/GTD). Clients who submit long dated orders must guarantee uniqueness across trading sessions. Note that the trading sessions for XSFE and NZFX span multiple days. Orders with duplicate ClOrdID identifiers will be rejected.
1	Account	String [10]	C	Up to 10 character account ID used for pre trade risk checks. Supplying more than 10 characters will cause the order to be rejected. Required by ASX for orders in XSFE and NZFX markets.
581	AccountType	int	C	Type of account associated with an order. Required to describe Account (1) value. Valid values: 1 = Customer (Client) 2 = Non Customer (House).



Tag	Name	Data Type	Reqd	Comment
55	Symbol	String [255]	C	Common, human understood representation of the security. Required if SecurityID (48) is not present. If SecurityID (48) is specified, set Symbol (55) to [N/A]. Order will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A].
48	SecurityID	String [10]	C	Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22). Symbol (55) must be set to [N/A] if SecurityID (48) is specified. The order will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A].
22	SecurityIDSource	String	C	Identifies the source of the SecurityID (48) value. Required if SecurityID (48) is specified. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
207	SecurityExchange	Exchange		Market of security. Valid values: XSFE = Sydney Futures Exchange NZFX = New Zealand Futures and Options Exchange.
54	Side	char	Y	Side of order. Valid values: 1 = Buy 2 = Sell.
38	OrderQty	Qty	Y	Total order quantity.



Tag	Name	Data Type	Reqd	Comment
40	OrdType	char	Y	Order type. Valid values: 2 = Limit K = Market to Limit.
44	Price	Price	C	Price per unit quantity. Required if OrdType (40) is Limit (2). Ignored if OrdType (40) is Market to Limit (K).
18	ExecInst	MultipleCharValue		Instructions for handling the order. Valid value: o = Cancel on connection loss. See 4.3 <i>Cancel on Disconnect</i> .
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> 448	PartyID	String [see comment]	C	Used to identify source of PartyID. Required if NoPartyIDs > 0. Required if PartyIDSource (447) is specified. The maximum number of characters depends on the PartyRole (452) value. See PartyRole (452).
> 447	PartyIDSource	char	C	Identifies class or source of PartyID (448) value. Required if PartyID (448) is specified. Valid values: D = Proprietary/Custom code.
> 452	PartyRole	int	C	The role of the party in the transaction.



Tag	Name	Data Type	Reqd	Comment
				<p>Required if PartyID is specified.</p> <p>Valid values:</p> <p>24 = Customer Account. Up to 15 characters. First 6 characters passed to clearing system - STP usage.</p> <p>45 = Secondary Account Number. Can be used as a secondary client order identifier. Up to 32 characters.</p> <p>76 = Desk ID. Identifies the SOG that this order is part of. Up to 10 characters.</p> <p>The following values are also accepted, however the usage is not supported or recommended on the current version of the derivatives platform:</p> <p>1 = Executing Firm. Ignored on entry, firm mnemonic set on messages from ASX.</p> <p>4 = Clearing Firm. Only clearing firms valid for the user are accepted.</p> <p>11 = Order Origination Trader. Accepted, but ignored. Not reflected in messages from ASX.</p> <p>12 = Executing Trader. The user within the member that owns order.</p> <p>29 = Intermediary. Accepted, but ignored. Not reflected in messages from ASX.</p> <p>36 = Entering Trader. Accepted and validated - only users in the entering firm are accepted. Usage not supported in derivatives.</p>
59	TimeInForce	char		<p>Specifies how long the order remains in effect. Absence of this field is interpreted as Day.</p> <p>Valid values:</p>



Tag	Name	Data Type	Reqd	Comment
				0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK) 6 = Good Till Date (GTD).
432	ExpireDate	LocalMktDate	C	Business date of order expiration (last day the order can trade); always expressed in terms of the local market date. Required if TimeInForce is GTD and ExpireTime is not specified. The order will be rejected if both ExpireDate (432) and ExpireTime (126) are specified.
126	ExpireTime	UTCTimestamp	C	Time of order expiration; always expressed in UTC. If milliseconds are specified in ExpireTime (126) they will be ignored. Required if TimeInForce is GTD and ExpireDate is not specified. The order will be rejected if ExpireDate (432) and ExpireTime (126) are both specified. A GTD order with ExpireTime (126) will expire at the earlier of these two times: the time specified in ExpireTime (126) or the close of the day session, when order book state transitions to maintenance state after close.
1624	NoMatchInst	NumInGroup		Number of instructions in the <MatchingInstructions> repeating group. Only 1 instruction is allowed. Additional instructions will be ignored.
> 1625	MatchInst	int	C	Matching instruction. Required if NoMatchInst > 0. Valid values:



Tag	Name	Data Type	Reqd	Comment
				1 = Match.
> 1626	MatchAttribTagID	TagNum	C	This tag is required by the FIX protocol when NoMatchInst > 0. Valid values: 10000 = Activate Unintentional Crossing protection (UCP) for this order.
> 1627	MatchAttribValue	int	C	Value to apply the matching instruction. Required if NoMatchInst > 0. When MatchAttribTagID=10000, the MatchAttribValue is a 32-bit signed integer number supplied as the UCP key to activate Unintentional Crossing Protection (UCP) for this order. If two orders from the same member or member unit with the same UCP keys match, the resultant trade will be handled as a private Booking Report. Public market data will reflect the resting order as being cancelled, or volume is decremented.
58	Text	String [128]		Free format text string. Value will be reflected in the Execution Report (8) message acknowledging the order.
60	TransactTime	UTCTimestamp	Y	Timestamp when the business transaction represented by the message occurred.
	StandardTrailer		Y	

4.4.1. NewOrderSingle (D) Example – SOG and Cancel on Disconnect

```
8=FIXT.1.1|9=161|35=D|49=ABCO1|56=ASX|34=25|52=20161123-22:57:48|11=ABC-D-2000|60=20161123-22:57:48|54=1|55=IRZ9|38=2|44=97.57|40=2|581=1|1=ABC1|453=1|448=2234|447=D|452=76|18=o|10=244|
```



```
8=FIXT.1.1|9=0000411|35=8|49=ASX|56=ABC01|34=25|52=20161123-22:57:48.333|369=25|37=6207326098104008705|
198=6207326098104008705|11=ABC-D-2000|453=6|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=2234|447=D|452=76|448=ABC01|447=D|452=36|17=6207326098104008705-1270|150=0|39=0|1=ABC1|
581=1|55=IRZ9|48=65017|22=8|54=1|38=2|40=2|44=97.57|15=AUD|59=0|18=0|151=2|14=0|6=0|60=20161123-22:57:48.330|
10=046|
```

4.4.2. *NewOrderSingle (D) Example – UCP Key Usage*

```
8=FIXT.1.1|9=167|35=D|49=ABC01|56=ASX|34=69|52=20161123-04:10:25|11=ABC-D-1|60=20161123-04:10:25|54=1|55=IRZ9|
38=10|44=97.55|40=2|59=0|581=1|1=ABC1|1624=1|1625=1|1626=10000|1627=35439|10=182|
```

```
8=FIXT.1.1|9=0000419|35=8|49=ASX|56=ABC01|34=211|52=20161123-04:10:25.221|369=69|37=6207042382328774657|
198=6207042382328774657|11=ABC-D-1|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|448=ASXCL|
447=D|452=4|448=ABC01|447=D|452=36|17=6207042382328774657-866|150=0|39=0|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|
38=10|40=2|44=97.55|15=AUD|59=0|151=10|14=0|6=0|60=20161123-04:10:25.218|1624=1|1625=1|1626=10000|1627=35439|
10=132|
```

4.4.3. *NewOrderSingle (D) Example – GTD Order, Specifying Date*

```
8=FIXT.1.1|9=144|35=D|49=ABC01|56=ASX|34=75|52=20161123-04:11:51|11=ABC-D-2|60=20161123-04:11:51|54=1|55=IRZ9|
38=10|44=97.55|40=2|59=6|581=1|1=ABC1|432=20161130|10=141|
```

```
8=FIXT.1.1|9=0000396|35=8|49=ASX|56=ABC01|34=217|52=20161123-04:11:51.173|369=75|37=6207042742833397761|
198=6207042742833397761|11=ABC-D-2|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|448=ASXCL|
447=D|452=4|448=ABC01|447=D|452=36|17=6207042742833397761-867|150=0|39=0|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|
38=10|40=2|44=97.55|15=AUD|59=6|432=20161130|151=10|14=0|6=0|60=20161123-04:11:51.169|10=112|
```



4.5. OrderCancelRequest (F)

The OrderCancelRequest message is used to cancel an order. Successful cancellation of an order is communicated by an ExecutionReport (8). If the cancel is rejected, an Order CancelReject (9) message is sent.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = F
37	OrderID	String [19]	C	Unique identifier for the order as assigned by ASX. Required, if OrigClOrdID (41) is not supplied. If supplied, OrigClOrdID (41) will be ignored.
11	ClOrdID	String [128]	Y	Unique identifier for the order as assigned by the client. Uniqueness must be guaranteed within a single trading session by the sender for clients who do not submit long dated orders (GTC/GTD). Clients who submit long dated orders must guarantee uniqueness across trading sessions. Note that the trading sessions for XSFE and NZFX span multiple days. Orders with duplicate ClOrdID identifiers will be rejected.
41	OrigClOrdID	String [128]	C	Last ClOrdID (11) used to successfully touch the order (NOT the initial order of the day) as assigned by the client on this FIX session. A ClOrdID (11) value entered in one FIX session cannot be used to control an order on another FIX session.



Tag	Name	Data Type	Reqd	Comment
54	Side	char	Y	Side of order. This tag is required to comply with the FIX protocol. The value is ignored.
38	OrderQty	Qty	Y	This tag is conditionally required to comply with the FIX protocol. The value is ignored.
55	Symbol	String [255]	C	Common, human understood representation of the security. This tag is conditionally required to comply with FIX protocol. The value is ignored. Required if SecurityID (48) is not present. If SecurityID (48) is specified, set Symbol (55) to [N/A]. Cancel will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A].
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22). Symbol (55) must be set to [N/A] if SecurityID (48) is specified. Cancel will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A]. This tag is conditionally required to comply with the FIX protocol. The value is ignored.



Tag	Name	Data Type	Reqd	Comment
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.
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> 448	PartyID	String [see comment]	C	Used to identify source of PartyID. Required if NoPartyIDs>0. Required if PartyIDSource (447) is specified. The maximum number of characters depends on the PartyRole (452) value. See PartyRole (452).
> 447	PartyIDSource	char	C	Identifies class or source of PartyID (448) value. Required if PartyID (448) is specified. Valid values: D = Proprietary/Custom code.
> 452	PartyRole	int	C	The role of the party in the transaction. Required if PartyID is specified. Valid values: 76 = Desk ID (Identifies the SOG that this order is part of. Required when cancelling a SOG order. Up to 10 characters).



Tag	Name	Data Type	Reqd	Comment
58	Text	String [128]	N	Free format text string.
60	TransactTime	UTCTimestamp	Y	Timestamp when the business transaction represented by the message occurred.
	StandardTrailer		Y	

4.5.1. OrderCancelRequest (F) Example – Basic Cancel

```
8=FIXT.1.1|9=138|35=F|49=ABCO1|56=ASX|34=100|52=20161123-04:17:54|11=ABC-F-20161123-04:17:54|
37=6207042382328774657|60=20161123-04:17:54|54=1|55=IRZ9|38=0|10=029|
```

```
8=FIXT.1.1|9=0000435|35=8|49=ASX|56=ABCO1|34=242|52=20161123-04:17:54.609|369=100|37=6207042382328774657|
198=6207042382328774657|11=ABC-F-20161123-04:17:54|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=6207042382328774657-868|150=4|39=4|1=ABC1|581=1|55=IRZ9|
48=65017|22=8|54=1|38=10|40=2|44=97.55|15=AUD|59=0|151=0|14=0|6=0|60=20161123-04:17:54.604|1624=1|1625=1|
1626=10000|1627=35439|10=211|
```

4.5.2. OrderCancelRequest (F) Example – SOG Cancel

This example shows a SOG order cancelled by a member of the SOG, different to the member who entered it.

Order cancelled by other user

```
8=FIXT.1.1|9=165|35=F|49=ABCO2|56=ASX|34=33|52=20161123-04:30:31|11=ABC-F-20161123-04:30:31|
37=6207047299407822849|60=20161123-04:30:31|54=1|55=IRZ9|38=0|453=1|448=2234|447=D|452=76|10=247|
```

Ack to cancelling user

```
8=FIXT.1.1|9=0000420|35=8|49=ASX|56=ABCO2|34=33|52=20161123-04:30:31.822|369=33|37=6207047299407822849|
198=6207047299407822849|11=ABC-F-20161123-04:30:31|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=6207047299407822849-4352|150=4|39=4|
1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=64|40=2|44=97.56|15=AUD|59=0|151=0|14=0|6=0|60=20161123-04:30:31.819|
10=036|
```

Ack to entering member

8=FIXT.1.1|9=0000421|35=8|49=ASX|56=ABC01|34=294|52=20161123-04:30:31.822|369=151|37=6207047299407822849|
198=6207047299407822849|11=ABC-F-20161123-04:30:31|453=6|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|
452=7|448=ASXCL|447=D|452=4|448=2234|447=D|452=76|448=ABC02|447=D|452=36|17=6207047299407822849-872|150=4|39=4|
1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=64|40=2|44=97.56|15=AUD|59=0|151=0|14=0|6=0|60=20161123-04:30:31.819|
10=097|



4.6. OrderCancelReplaceRequest (G)

The OrderCancelReplaceRequest is used to change the parameters of an existing order. Not all parameters can be modified. The message definition states when a parameter must be specified to comply with the FIX protocol, but cannot be changed.

- A successful modification will result in an ExecutionReport (8).
- Where the OrderCancelReplaceRequest makes one of the following changes:
 - modifies price
 - increases quantity
 - modifies AccountType (581)

the order price-time priority will be affected. A new public order identifier, SecondaryOrderID (198), will be issued in the ExecutionReport (8).

- Where the request modifies the quantity down, then the order price-time priority will remain.
- An unsuccessful modification will be rejected with an OrderCancelReject (9).
- To cancel an order, the OrderCancelRequest (F) is used.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = G
37	OrderID	String [19]	C	Unique identifier for the order as assigned by ASX. Required if OrigClOrdID (41) is not supplied. If supplied, OrigClOrdID (41) will be ignored.
11	ClOrdID	String [128]	Y	Unique identifier for the order as assigned by the client. Uniqueness must be guaranteed within a single trading session by the sender for clients who do not submit long dated orders (GTC/GTD). Clients submitting long dated orders must guarantee uniqueness of all orders across trading sessions and business trading dates.



Tag	Name	Data Type	Reqd	Comment
				Replacing an order with a duplicate ClOrdID identifier will be rejected.
41	OrigClOrdID	String [128]	C	<p>Last ClOrdID (11) used to successfully touch the order (NOT the initial order of the day) as assigned by the client on this FIX session.</p> <p>Required if OrderID (37) is not supplied. Used only when OrderID (37) is not present.</p> <p>A ClOrdID (11) value entered in one FIX session cannot be used to control an order on another FIX session.</p>
1	Account	String [10]	C	<p>10 character account ID used for pre trade risk checks. Supplying more than 10 characters will cause the order to be rejected.</p> <p>Required by ASX for orders in XSFE and NZFX markets.</p>
581	AccountType	int	C	<p>Type of account associated with an order. Required to describe Account (1) value.</p> <p>Valid values:</p> <p>1 = Customer</p> <p>2 = Non-customer.</p>
55	Symbol	String [255]	C	<p>Common, human understood representation of the security. This tag is conditionally required to comply with the FIX protocol. The value is ignored.</p> <p>Required if SecurityID (48) is not present. If SecurityID (48) is specified, set Symbol (55) to [N/A].</p> <p>Replace will be rejected, if SecurityID (48) is present and Symbol (55) is not [N/A].</p>



Tag	Name	Data Type	Reqd	Comment
48	SecurityID	String [10]	C	Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22). Symbol (55) must be set to [N/A] if SecurityID (48) is specified. Replace will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A]. This tag is conditionally required to comply with the FIX protocol. The value is ignored.
22	SecurityIDSource	String	C	Identifies the source of the SecurityID (48) value. Required if SecurityID (48) is specified. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID. This tag is conditionally required to comply with the FIX protocol. The value is ignored.
207	SecurityExchange	Exchange		Market of security. Valid values: XSFE = Sydney Futures Exchange NZFX = New Zealand Futures and Options Exchange.
54	Side	char	Y	Side of order. Must be the same value as on the original order. This tag is conditionally required to comply with FIX protocol. Valid values: 1 = Buy 2 = Sell.
38	OrderQty	Qty	Y	Total order quantity.



Tag	Name	Data Type	Reqd	Comment
40	OrdType	char	Y	Order type. Valid values: 2 = Limit K = Market with Left Over as Limit (market order with unexecuted quantity becoming limit order at last price).
44	Price	Price	C	Price per unit quantity. Required if OrdType (40) is Limit (2). Ignored if OrdType (40) is Market to Limit (K).
18	ExecInst	MultipleCharValue		Instructions for handling the order. Replacement order must be created with new parameters (that is, original order values will not be brought forward to replacement order unless redefined within this message). Valid values: o = Cancel on connection loss. See 4.3 <i>Cancel on Disconnect</i> .
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> 448	PartyID	String	C	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).



Tag	Name	Data Type	Reqd	Comment
> 447	PartyIDSource	char	C	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Valid values: D = Proprietary/custom code.
> 452	PartyRole	int	C	The role of the party in the transaction. Required if PartyID is specified. Valid values: 24 = Customer Account (Up to 15 characters. First 6 characters passed to clearing system - STP usage. Modifications will only affect trades subsequent to this update) 45 = Secondary Account Number (Modifications will only affect trades subsequent to this update. Up to 32 characters) 76 = Desk ID (SOG ID. Must be specified on modification if on order, otherwise modification will be rejected. Cannot be changed to a new SOG. Up to 10 characters.)
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) 6 = Good Till Date (GTD).



Tag	Name	Data Type	Reqd	Comment
432	ExpireDate	LocalMktDate	C	<p>Business date of order expiration (last day the order can trade), always expressed in terms of the local market date.</p> <p>Required if TimeInForce is GTD and ExpireTime is not specified.</p> <p>Can only be modified if previously specified on the order, or the order does not contain ExpireTime (126).</p> <p>Replace will be rejected if ExpireDate (432) and ExpireTime (126) are specified.</p>
126	ExpireTime	UTCTimestamp	C	<p>Time/Date of order expiration; always expressed in UTC. If milliseconds are specified in ExpireTime (126) they will be ignored.</p> <p>Required if TimeInForce is GTD and ExpireDate is not specified.</p> <p>Can only be modified if previously specified on the order, or the order does not contain ExpireDate (432).</p> <p>The replace will be rejected if ExpireDate (432) and ExpireTime (126) are both specified.</p> <p>A GTD order with ExpireTime (126) will expire at the earlier of these two times: the time specified in ExpireTime (126) or the close of the day session, when order book state transitions to maintenance state after close.</p>
1624	NoMatchInst	NumInGroup		<p>Number of instructions in the <MatchingInstructions> repeating group. Only 1 instruction is allowed. Replacement order must be created with new parameters (that is, original order values will not be brought forward to replacement order unless redefined within this message).</p>
> 1625	MatchInst	int	C	<p>Matching instruction. Required if NoMatchInst > 0.</p>



Tag	Name	Data Type	Reqd	Comment
				Valid values: 1 = Match.
> 1626	MatchAttribTagID	TagNum	C	This tag is required by the FIX protocol when NoMatchInst > 0. Valid values: 10000 = UCP Key.
> 1627	MatchAttribValue	int	C	Value to apply the matching instruction. Required if NoMatchInst > 0. When MatchAttribTagID=10000, the MatchAttribValue is a 32-bit signed integer number supplied as the UCP key to activate Unintentional Crossing Protection (UCP) for this order. If two orders from the same member or member unit with the same UCP keys match, the resultant trade will be handled as a private Booking Report. Public market data will reflect the resting order as being cancelled, or volume is decremented.
58	Text	String [128]		Free format text string. Value will be reflected in the Execution Report (8) message, acknowledging the modification.
60	TransactTime	UTCTimestamp	Y	Timestamp when the business transaction represented by the message occurred.
	StandardTrailer		Y	



4.6.1. OrderCancelReplaceRequest (G) Example – Amend of TIF and Quantity

```
8=FIXT.1.1|9=170|35=G|49=ABCO1|56=ASX|34=171|52=20161123-04:35:19|11=ABC-G-20161123-04:35:19|60=20161123-04:35:19|54=1|55=IRZ9|38=9|44=97.55|40=2|59=1|37=6207042742833397761|1=ABC1|581=1|10=255|  
8=FIXT.1.1|9=0000398|35=8|49=ASX|56=ABCO1|34=314|52=20161123-04:35:19.359|369=171|37=6207042742833397761|198=6207042742833397761|11=ABC-G-20161123-04:35:19|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=6207042742833397761-873|150=5|39=0|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=9|40=2|44=97.55|15=AUD|59=1|151=9|14=0|6=0|60=20161123-04:35:19.350|10=047|
```

4.6.2. OrderCancelReplaceRequest (G) Example – SOG Amend

This example shows a SOG order amended by a member of the SOG, different to the member who entered it.

Order amended by other user

```
8=FIXT.1.1|9=197|35=G|49=ABCO2|56=ASX|34=79|52=20161123-04:41:24|11=ABC-G-20161123-04:41:24|60=20161123-04:41:24|54=1|55=IRZ9|38=3|44=97.55|40=2|59=0|37=6207049519322906625|453=1|448=2234|447=D|452=76|1=ABC1|581=1|10=220|
```

Ack to amending user

```
8=FIXT.1.1|9=0000419|35=8|49=ASX|56=ABCO2|34=78|52=20161123-04:41:24.475|369=79|37=6207049519322906625|198=6207050180601069569|11=ABC-G-20161123-04:41:24|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=6207049519322906625-4354|150=5|39=0|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=3|40=2|44=97.55|15=AUD|59=0|151=3|14=0|6=0|60=20161123-04:41:24.471|10=233|
```

Ack to entering member

```
8=FIXT.1.1|9=0000420|35=8|49=ASX|56=ABCO1|34=340|52=20161123-04:41:24.475|369=196|37=6207049519322906625|198=6207050180601069569|11=ABC-G-20161123-04:41:24|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=6207049519322906625-876|150=5|39=0|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=3|40=2|44=97.55|15=AUD|59=0|151=3|14=0|6=0|60=20161123-04:41:24.471|10=013|
```

4.7. ExecutionReport (8)

The Execution Report (8) message is used to:

- Confirm the receipt of an order
- Confirm changes to an existing order
- Confirm or convey an order cancellation or expiration
- Convey fill information on working orders
- Reject orders
- Convey information about re-stated long orders carried from one trading session to the next.

Prices are formatted to contain the number of decimal places for the relevant instrument, as distributed in reference data. For prices that relate to the main (head) order, the relevant instrument is the order instrument. For prices that relate to legs, the relevant instrument is the one for that leg.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 8
37	OrderID	String [19]	Y	Unique identifier for Order as assigned by ASX. Private to the member firm entering the order. Returned as zero for execution reports notifying of the following conditions: <ul style="list-style-type: none">• the bust of a leg trade of a spread order• failure to enter a new order (e.g. invalid quantity)• no orders matching an OrderMassStatusRequest (AF).
198	SecondaryOrderID	String [19]		Unique identifier for Order as assigned by ASX. Provided on public market data.
11	ClOrdID	String [128]		Unique identifier for Order as assigned by the client.



Tag	Name	Data Type	Reqd	Comment
41	OrigClOrdID	String [128]		ClOrdID (11) of the previous order. Provided in responses to a Cancel or Cancel/Replace request, if provided by the user of this FIX connection in the Cancel or Cancel/Replace message.
584	MassStatusReqID	String [64]		Value assigned by issuer of Mass Status Request to uniquely identify the request. Supplied on Execution Report (8) messages in response to an OrderMassStatusRequest (AF).
911	TotNumReports	int		Total number of reports returned in response to a request. Supplied on Execution Report (8) messages in response to an OrderMassStatusRequest (AF).
912	LastRptRequested	Boolean		Indicates whether this message is the last report message in response to a request message, for example, OrderMassStatusRequest(35=AF). Valid values: N = Not the last message Y = Last message.



Tag	Name	Data Type	Reqd	Comment
18	ExecInst	MultipleCharValue		<p>Instructions for handling the order.</p> <p>Valid values:</p> <ul style="list-style-type: none"> o = Cancel on connection loss. <p>If an order marked as Cancel on connection loss is cancelled by the system due to a disconnect or logout, the execution report is available when connection is re-established. See 3.3 <i>Lost connection</i>.</p>
453	NoPartyIDs	NumInGroup		Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries.
> 448	PartyID	String [see comment]		<p>Identifies the party. Required if NoPartyIDs > 0. Required if PartyIDSource is specified.</p> <p>The maximum number of characters depends on the PartyRole (452) value. See PartyRole (452).</p>
> 447	PartyIDSource	char		<p>Identifies class or source of the PartyID (448) value. Required if PartyID is specified.</p> <p>Valid values:</p> <ul style="list-style-type: none"> D = Proprietary/Custom code.

Tag	Name	Data Type	Reqd	Comment
> 452	PartyRole	int		<p>The role of the party in the transaction. Required if PartyID is specified. Maximum number of characters for each type in [].</p> <p>Valid values:</p> <ul style="list-style-type: none"> 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 SOG that this order is part of) [10].
1057	AggressorIndicator	Boolean		<p>Used to identify whether the order initiator is an aggressor or not in the trade.</p> <p>Valid values:</p> <ul style="list-style-type: none"> Y = Order initiator is aggressor N = Order initiator is passive. <p>If tag is not defined, both orders are passive such as in auctions.</p>



Tag	Name	Data Type	Reqd	Comment
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.
> 1625	MatchInst	int		<p>Matching instruction. Supplied if NoMatchInst > 0</p> <p>Valid values:</p> <p>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.</p> <p>2 = Do Not Match: Only present where ExecType=Trade (150=F) and UCP keys did not match for this trade*</p> <p>* 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.</p>
> 1626	MatchAttribTagID	TagNum		<p>This tag is required by the FIX protocol when NoMatchInst > 0.</p> <p>Valid values:</p> <p>10000 = UCP Key.</p>
> 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).



Tag	Name	Data Type	Reqd	Comment
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.
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. When ExecType=Rejected (150=8), it describes the error in human readable format. The maximum number of characters may exceed 128 for complex error messages.
17	ExecID	String [84]	Y	Unique identifier of execution message as assigned by the ASX. When ExecType=Order Status (150=I) ExecID will be zero (17=0).
19	ExecRefID	String [84]		Execution reference identifier. Supplied when ExecType=Trade Cancel (150=H).
150	ExecType	char	Y	Describes the specific ExecutionReport (8), for example 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 8 = Rejected C = Expired D = Restated



Tag	Name	Data Type	Reqd	Comment
				F = Trade (partial fill or fill) H = Trade Cancel I = Order Status.
39	OrdStatus	char	Y	Identifies status of order. Pending Cancel/Pending Replace may be sent when an order is being reported upon, while a cancel or replace has been received but not processed. Valid values: 0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Cancelled 6 = Pending Cancel 8 = Rejected C = Expired E = Pending Replace.
103	OrdRejReason	int		Code to identify reason for order rejection. Provided when an order is rejected. Valid values: 0 = Broker/Exchange Option (Used in several situations: Invalid shared order group, order books for one or more legs in an inter-calendar or UDC contract are closed) 1 = Unknown symbol 2 = Exchange closed (order book state prevents order entry) 3 = Order exceeds limit (PTRM or TPL limit exceeded)



Tag	Name	Data Type	Reqd	Comment
				4 = Too late to enter 5 = Unknown order 6 = Duplicate order 11 = Unsupported order characteristic 13 = Incorrect quantity 15 = Unknown account(s); user does not have access to PTRM account 18 = Invalid price increment. 99 = Other.
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.
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.



Tag	Name	Data Type	Reqd	Comment
				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
55	Symbol	String [255]		Common, human understood representation of the security. Value as entered by the user, or determined from SecurityID (48).
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type. Value as entered by the user, or determined from Symbol (55).
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.
29	LastCapacity	char		Used to indicate capacity of broker. Valid values for XSFE and NZFX: 3 = Cross as principal.



Tag	Name	Data Type	Reqd	Comment
54	Side	char	Y	Side of order. Valid values: 1 = Buy 2 = Sell 7 = Undisclosed. Response to OrderMassStatusRequest to indicate there are no open orders.
38	OrderQty	Qty	Y	Total order quantity.
880	TrdMatchID	String [10]		Identifier assigned to a trade for ASX clearing purposes – clearing deal number. For the 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).
40	OrdType	char		Order type. Valid values: 2 = Limit K = Market to Limit.
44	Price	Price		Price per unit quantity entered on order.
15	Currency	Currency		Identifies currency used for price.



Tag	Name	Data Type	Reqd	Comment
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).
32	LastQty	Qty		Quantity bought/sold on this (last) fill. Supplied if ExecType=Trade (150=F).
31	LastPx	Price		Price of this (last) fill. Supplied if ExecType=Trade (150=F).
151	LeavesQty	Qty	Y	Quantity open for further execution. Set to zero for cancelled and rejected orders.
14	CumQty	Qty	Y	Total quantity filled of all executions for the order. On execution, reports reporting the bust of a trade, originally reported as a leg trade of spread order, CumQty is set to zero.
6	AvgPx	Price		Calculated average price of all fills on this order. Ignore for products quoted in Yield.
75	TradeDate	LocalMktDate		Business trade date. Supplied if ExecType is Fill or Trade Cancel (150=F or H). Also supplied if ExecType is Order Status (150=I) when there has been a trade for the order in the current trading session.
60	TransactTime	UTCTimestamp		Timestamp when the business transaction, represented by the message, occurred.



Tag	Name	Data Type	Reqd	Comment
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		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 trade identifier assigned to this leg match.
> 1366	LegAllocID	String [10]		Identifier assigned to a trade for ASX clearing purposes - clearing deal number.
> 566	LegPrice	Price		Price for leg of a multileg (only present for absolute pricing).
> 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. Valid values: 1 = Buy 2 = Sell.
> 556	LegCurrency	Currency		Multileg instrument's currency.
> 685	LegOrderQty	Qty		Quantity ordered of this leg.



Tag	Name	Data Type	Reqd	Comment
> 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	



4.7.1. ExecutionReport (8) Example – Futures Contract, Market to Limit with UCP Key Applied

This example shows a market-to-limit order with a UCP key applied. There are executions against orders from the same member firm with and without UCP key applied. There are also executions against other member firms.

Order entry

```
8=FIXT.1.1|9=164|35=D|49=ABCO1|56=ASX|34=99|52=20161123-23:16:26|11=ABC-D-2001|60=20161123-23:16:26|54=1|55=APH7|38=100|40=K|581=1|1=ABC1|1624=1|1625=1|1626=10000|1627=5924677|18=o|10=127|
```

Ack - note OrdType (40)

```
8=FIXT.1.1|9=0000431|35=8|49=ASX|56=ABCO1|34=100|52=20161123-23:16:26.713|369=99|37=6207330788929716225|198=6207330788929716225|11=ABC-D-2001|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=6207330788929716225-1320|150=0|39=0|1=ABC1|581=1|55=APH7|48=58950|22=8|54=1|38=100|40=2|44=5323|15=AUD|59=0|18=o|151=100|14=0|6=0|60=20161123-23:16:26.710|1624=1|1625=1|1626=10000|1627=5924677|10=230|
```

Fill on entry – different member firm is counterparty

```
8=FIXT.1.1|9=0000483|35=8|49=ASX|56=ABCO1|34=101|52=20161123-23:16:26.714|369=99|37=6207330788929716225|198=6207330788929716225|11=ABC-D-2001|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=100000001|17=6207330788929716225-B|150=F|39=1|1=ABC1|581=1|574=4|55=APH7|48=58950|22=8|54=1|38=100|40=2|44=5323|15=AUD|59=0|18=o|1057=Y|32=5|31=5323|151=95|14=5|6=5323|75=20161124|60=20161123-23:16:26.710|1624=1|1625=2|1626=10000|1627=5924677|10=186|
```

Fill, own firm as counterparty, UCP key set on other order. Note party tags for contra side

```
8=FIXT.1.1|9=0000512|35=8|49=ASX|56=ABCO1|34=107|52=20161123-23:17:50.214|369=104|37=6207330788929716225|198=6207330788929716225|11=ABC-D-2001|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|880=100000002|17=6207331139154100225-B|150=F|39=1|1=ABC1|581=1|574=4|55=APH7|48=58950|22=8|54=1|38=100|40=2|44=5323|15=AUD|59=0|18=o|1057=N|32=10|31=5323|29=3|151=85|14=15|6=5323|75=20161124|60=20161123-23:17:50.210|1624=1|1625=1|1626=10000|1627=5924677|10=227|
```



Fill, own firm as counterparty, UCP not key set on other order. Note party tags for contra side

```
8=FIXT.1.1|9=0000512|35=8|49=ASX|56=ABC01|34=109|52=20161123-23:18:18.473|369=106|37=6207330788929716225|
198=6207330788929716225|11=ABC-D-2001|453=7|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ABC|447=D|452=17|448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|880=1000000003|17=6207331257685131265-B|150=F|
39=1|1=ABC1|581=1|574=4|55=APH7|48=58950|22=8|54=1|38=100|40=2|44=5323|15=AUD|59=0|18=o|1057=N|32=15|31=5323|29=3|
151=70|14=30|6=5323|75=20161124|60=20161123-23:18:18.470|1624=1|1625=2|1626=10000|1627=5924677|10=016|
```

Fully filled against different counterparty

```
8=FIXT.1.1|9=0000486|35=8|49=ASX|56=ABC01|34=111|52=20161123-23:18:35.911|369=108|37=6207330788929716225|
198=6207330788929716225|11=ABC-D-2001|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|880=1000000004|17=6207331330817015809-B|150=F|39=2|1=ABC1|581=1|
574=4|55=APH7|48=58950|22=8|54=1|38=100|40=2|44=5323|15=AUD|59=0|18=o|1057=N|32=70|31=5323|151=0|14=100|6=5323|
75=20161124|60=20161123-23:18:35.906|1624=1|1625=2|1626=10000|1627=5924677|10=036|
```

4.7.2. ExecutionReport (8) Example – Spread Contract

This example shows a spread contract acknowledged and filled. Note the legs on the fill.

Order Entry

```
8=FIXT.1.1|9=131|35=D|49=ABC01|56=ASX|34=278|52=20161123-05:01:02|11=ABC-D-10|60=20161123-05:01:02|54=1|55=IRZ0U1|
38=2|44=.5|40=2|59=0|581=1|1=ABC1|10=035|
```

Ack

```
8=FIXT.1.1|9=0000383|35=8|49=ASX|56=ABC01|34=422|52=20161123-05:01:02.548|369=278|37=6207055121784782849|
198=6207055121784782849|11=ABC-D-10|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|17=6207055121784782849-930|150=0|39=0|1=ABC1|581=1|55=IRZ0U1|
48=77961|22=8|54=1|38=2|40=2|44=0.5|15=AUD|59=0|151=2|14=0|6=0|60=20161123-05:01:02.541|10=022|
```



Fill

```
8=FIXT.1.1|9=0000651|35=8|49=ASX|56=ABC01|34=425|52=20161123-05:01:40.158|369=280|37=6207055121784782849|
198=6207055121784782849|11=ABC-D-10|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|880=1000000007|17=6207055279398338561-B|150=F|39=2|1=ABC1|581=1|
574=R|55=IRZ0U1|48=77961|22=8|54=1|38=2|40=2|44=0.5|15=AUD|59=0|1057=N|32=2|31=0.5|151=0|14=2|6=0.5|75=20161123|
60=20161123-05:01:40.119|555=2|600=IRU1|602=77937|603=8|1788=6207055279398338563-A|624=2|556=AUD|685=2|
1366=1000000009|637=97.47|1418=2|600=IRZ0|602=65018|603=8|1788=6207055279398338562-B|624=1|556=AUD|685=2|
1366=1000000008|637=97.97|1418=2|10=110|
```

4.7.3. ExecutionReport (8) Example – Order Expires Due to TIF Settings

```
8=FIXT.1.1|9=0000411|35=8|49=ASX|56=ABC01|34=793|52=20161123-06:15:16.889|369=603|37=6207073641532768257|
198=6207073641532768257|11=ABC-D-39|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|17=6207073641532768257-1172|150=C|39=C|1=ABC1|581=1|55=IRZ9|48=65017|
22=8|54=1|38=10|40=2|44=97.55|15=AUD|59=6|126=20161123-06:15:16.000|151=0|14=0|6=0|60=20161123-06:15:16.886|
10=165|
```

4.7.4. ExecutionReport (8) Example – Restatement

```
8=FIXT.1.1|9=0000389|35=8|49=ASX|56=ABC01|34=723|52=20161123-05:58:00.247|369=533|37=6207062394724696065|
198=6207062394724696065|11=ABC-D-35|453=5|448=ABC|447=D|452=1|448=ABC01|447=D|452=12|448=ABC|447=D|452=7|
448=ASXCL|447=D|452=4|448=ABC01|447=D|452=36|17=6207062394724696065-1128|150=D|39=0|378=1|1=ABC1|581=1|55=IRZ9|
48=65017|22=8|54=1|38=84|40=2|44=95|15=AUD|59=1|151=84|14=0|6=0|60=20161123-05:58:00.230|10=067|
```

4.7.5. Expiry of Orders

When an order expires, an execution report will be sent. The type of execution report depends on whether the order expired due to instrument expiry or order age:

- Age of order: ExecType=Expired (150=C).



- Instrument expiry: ExecType=Cancelled (150=4)

The time of expiry depends on the type of order and the contract.

Order Type	FIX Tags	When Issued
Day Order	TimelnForce=Day (59=0)	End of trading session: <ul style="list-style-type: none"> • Outright contracts and intra-spreads and 1:1 inter-spreads: when order book state transitions to the first maintenance state after close • All other contracts: when order book state transitions to close.
GTD timed order	TimelnForce=Good till date (59=6) ExpireTime=<expiry time> (126=<expiry time>)	At the time specified, subject to time handling for timed orders. See ExpireTime (126).
GTD dated order	TimelnForce=Good till date (59=6) ExpireDate=<expiry date> (432=<expiry date>)	The earliest of either: <ul style="list-style-type: none"> • End of business trade date specified in ExpireDate (432), when order book state transitions to the first maintenance state after close • At time of contract expiry, when order book state transitions to the first maintenance state after close.
GTC orders	TimelnForce=Good till cancel (59=1)	End of trading session when order book state transitions to the first maintenance state after close.
All active orders, early contract expiry	N/A	<ul style="list-style-type: none"> • Outright contracts: when order book state transitions to the first maintenance state after close • All other contracts: when order book state transitions to close.



4.7.6. Restatement of Long Orders

Additional messages are sent to notify the status of long orders (GTC and dated GTD orders).

- During maintenance, after close: Execution Report (8) with ExecType=Done for Day (150=3)
- During maintenance, before pre-open: Execution Report (8) with ExecType=Restated (150=D) and ExecRestatementReason=GT renewal/restatement (378=1)

If PTRM trade limits are reduced prior to orders being restored, orders that breach the lower PTRM limits will be cancelled.



4.8. OrderCancelReject (9)

The OrderCancelReject message is issued by ASX upon receipt of an OrderCancelRequest (F) or OrderCancelReplaceRequest (G) message, which cannot be honoured.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = 9
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 in response to an attempt to update or cancel an unknown order.
198	SecondaryOrderID	String [19]		Unique identifier for the order as assigned by ASX. Provided on public market data.
11	ClOrdID	String [128]	Y	ClOrdID (11) supplied by client on OrderCancelRequest (F) or OrderCancelReplaceRequest (G) message.
41	OrigClOrdID	String [128]		OrigClOrdID (41) supplied by client on OrderCancelRequest (F) or OrderCancelReplaceRequest (G) message.
39	OrdStatus	char	Y	Identifies status of order. Valid values: 0 = New 1 = Partially filled 2 = Filled 3 = Done for Day 4 = Cancelled 8 = Rejected C = Expired.



Tag	Name	Data Type	Reqd	Comment
				<p>If CxlRejReason = Unknown order (102 = 1); returns Rejected (8). If the OrderCancelReject (9) message is issued because an order has been fully filled or cancelled, there is a special case where the ExecutionReport (8) message advising of the full fill or cancel is sent to the FIX client after the OrderCancelReject message. In this case, the OrderCancelReject message will contain an OrdStatus (39) value representing the previous state of the order, not the current state. CxlRejReason will be Too late to cancel (102=0) and Text (58) will indicate that current OrdStatus (39) is unknown.</p>
434	CxlRejResponseTo	char	Y	<p>Identifies the type of request that a Cancel Reject is in response to.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 1 = Order cancel request 2 = Order cancel/replace request.
102	CxlRejReason	int		<p>Code to identify reason for cancel rejection.</p> <p>Valid values:</p> <ul style="list-style-type: none"> 0 = Too late to cancel 1 = Unknown order 2 = Broker/Exchange Option 6 = Duplicate CLOrdID (11) received 8 = Price exceeds current price band 100 = Order not currently accessible for cancellation or modification



Tag	Name	Data Type	Reqd	Comment
58	Text	String [see comment]		Free format text string that describes the error in human readable format. The maximum number of characters may exceed 128 for complex error messages.
60	TransactTime	UTCTimestamp	Y	Timestamp when the business transaction, represented by the message, occurred.
	StandardTrailer			

4.8.1. OrderCancelReject (9) Example

```
8=FIXT.1.1|9=140|35=F|49=ABC01|56=ASX|34=304|52=20161123-05:06:58|11=ABC-F-20161123-05:06:58|
37=6207055884468633601|60=20161123-05:06:58|54=1|55=IRZ0U1|38=0|10=161|
```

```
8=FIXT.1.1|9=0000175|35=9|49=ASX|56=ABC01|34=448|52=20161123-05:06:58.700|369=304|37=6207055884468633601|11=ABC-F-
20161123-05:06:58|39=4|60=20161123-05:06:58.700|434=1|102=0|58=Too late to cancel|10=023|
```



4.9. QuoteRequest (R)

The Quote Request message is used to broadcast an RFQ to the market. The Quote Request can be for a Bid, Offer, Two-Sided Quote, or for a Crossing. The Crossing Quote Request should be used to initiate Pre-Neg trades.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = R
131	QuoteReqID	String [64]	Y	Unique identifier for a QuoteRequest (35=R).
146	NoRelatedSym	NumInGroup	Y	Number of related symbols (instruments) in this request.
> 55	Symbol	String [255]	C	Common, human understood representation of the security. Required if SecurityID (48) is not present. Quote will be rejected if both Symbol (55) and SecurityID (48) are present.
> 48	SecurityID	String [10]	C	Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22). Required if Symbol (55) is not present. Quote will be rejected if SecurityID (48) is present and Symbol (55) is not [N/A].
> 22	SecurityIDSource	String	C	Identifies the source of the SecurityID (48) value. Required if SecurityID (48) is specified. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
> 54	Side	char		Side of order – where this Tag is not defined, Tag indicates a two-sided quote.



Tag	Name	Data Type	Reqd	Comment
				Valid values: 1 = Buy 2 = Sell 8 = Crossing.
> 38	OrderQty	Qty		OrderQty requested for RFQ.
	StandardTrailer		Y	

4.9.1. QuoteRequest (R) Example

8=FIXT.1.1|9=101|35=R|49=ABC01|56=ASX|34=315|52=20161123-05:09:42|131=ABC-R-12|146=1|55=[N/A]|48=77961|22=8|54=1|38=2|10=179|



4.10. QuoteRequestReject (AG)

The QuoteRequestReject (AG) message is used to inform a trading participant that their QuoteRequest (R) message has been rejected.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = AG
131	QuoteReqID	String [64]	Y	Identifier supplied on QuoteRequest (R) message being rejected.
658	QuoteRequestRejectReason	int	Y	Reason the QuoteRequest (R) was rejected. Valid values: 99 = Other - Text (58) will contain more specific information.
146	NoRelatedSym	NumInGroup	Y	Number of related symbols (instruments) in Request.
> 55	Symbol	String [255]	Y	Common, human understood representation of the security.
> 48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22).
> 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 Valid values: 1 = Buy 2 = Sell 8 = Crossing.
> 38	OrderQty	Qty	Y	Total quantity



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

4.10.1. QuoteRequestReject (AG) Example

8=FIXT.1.1|9=102|35=R|49=ABCO1|56=ASX|34=322|52=20161123-05:11:08|131=ABC-R-13|146=1|55=[N/A]|48=779611|22=8|54=1|38=2|10=223|

8=FIXT.1.1|9=0000144|35=AG|49=ASX|56=ABCO1|34=465|52=20161123-05:11:08.571|369=322|131=ABC-R-13|658=99|146=1|55=[N/A]|48=779611|22=8|54=1|38=2|58=Unknown instrument|10=044|



4.11. SecurityDefinitionRequest (c)

The SecurityDefinitionRequest (c) message is used to request the creation of a User Defined Combination (UDC) instrument.

To create a UDC with all legs having the same side, specify the leg side as buy. If the leg side is sell, the request will be rejected and a UDC with all buy legs (the inverse) will be created.

If the desired UDC has a fixed buy leg and all other legs as sells, the request will be rejected and a UDC with the fixed leg as sell and other legs as buy legs (the inverse) will be created. See 4.12 SecurityDefinition (d).

In response, a SecurityDefinition (d) message will be sent to confirm or reject the creation of the instrument. In addition, if a UDC was created, the definition with all the legs will be sent in a SecurityList (y) message, disseminated via FIX Market Data.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = c (lowercase)
320	SecurityReqID	String [64]	Y	Unique client generated ID of the Security Definition Request.
321	SecurityRequestType	int	Y	Type of Security Definition Request. Valid values: 1 = Request Security identity for the specifications provided (name of the security is not supplied).
167	SecurityType	String	C	Indicates the type of security requested. Required by ASX. Valid values: MLEG = Multileg Instrument.
55	Symbol	String [255]	C	Required by ASX. Set to [N/A].
1378	MultilegPriceMethod	int	C	Code to represent how the multileg price is to be interpreted when applied to the legs. Required by ASX.



Tag	Name	Data Type	Reqd	Comment
				Valid values: 0 = Net Price 3 = Individual (absolute price).
555	NoLegs	NumInGroup	C	Number of legs that make up the User Defined Combination. 2 - 6 legs are supported.
> 600	LegSymbol	String [255]	C	Common, human understood representation of the security. Required if LegSecurityID (602) is not present. If LegSecurityID (602) is specified, set LegSymbol (600) to [N/A]. Message will be rejected if LegSecurityID (602) is present and LegSymbol (600) is not [N/A].
> 602	LegSecurityID	String [10]	C	Security identifier value of LegSecurityIDSource (603) type. Requires LegSecurityIDSource (603). LegSymbol (600) must be set to [N/A], if LegSecurityID (602) is specified. Message will be rejected if LegSecurityID (602) is present and LegSymbol (600) is not [N/A].
> 603	LegSecurityIDSource	String	C	Identifies the source of the LegSecurityID (602) value. Required if LegSecurityID (602) is specified. Valid values: 8 = Exchange defined. Available as Tradeable Instrument ID.
> 623	LegRatioQty	float	C	Defines the volume an order with volume 1 in the combination book represents in this leg order book. ASX accepts only whole numbers for ratios.



Tag	Name	Data Type	Reqd	Comment
> 624	LegSide	char	C	The side of this individual leg of the multileg security. Specifies if this leg is buy or sell when the combination order it belongs to is buy. Valid values: 1 = Buy 2 = Sell.
> 566	LegPrice	Price	C	Price for leg of multileg. Provide on one leg only. Required if MultilegPriceMethod=Individual (1378 = 3).
	StandardTrailer		Y	

4.11.1. SecurityDefinitionRequest (c) Example

```
8=FIXT.1.1|9=156|35=c|49=ABC01|56=ASX|34=331|52=20161123-05:13:08|320=ABC-c-20161123-05:13:08|321=1|167=MLEG|
1378=0|555=2|600=IRH7|624=1|623=1|600=IRM8|624=2|623=2|55=[N/A]|10=238|
8=FIXT.1.1|9=0000156|35=d|49=ASX|56=ABC01|34=474|52=20161123-05:13:09.043|369=331|320=ABC-c-20161123-05:13:08|
322=RABC-c-20161123-05:13:08|323=1|55=UDC_IR-F_10MER|48=82157|22=8|10=069|
```



4.12. SecurityDefinition (d)

The Security Definition (d) message is used to confirm the acceptance or rejection of a request to create a User Defined Combination (UDC) from a Security Definition Request (c) message.

If a request to create a UDC was rejected because the UDC (or an inverse of it) exists in the system, the response will provide details of the existing UDC. For example, a request for a UDC with all sell legs will be created with all buy legs.

In addition, if a UDC was successfully created, the definition with all the legs will be sent in a SecurityList (y) message, disseminated via FIX Market Data.

4.12.1. Sorting of legs in UDCs

If the security proposal is accepted, but the legs are re-sorted (SecurityResponseType=Accept with revisions, 323=2), the UDC legs are sorted according to the following steps:

1. If an absolute price UDC (MultigetPriceMethod=Individual, 1378=3), the fixed-price leg is placed first.
2. Remaining legs are sorted alphabetically, by product (e.g. IR, XT).
3. Within each product, futures are placed before options.
4. Futures are sorted by maturity date.
5. Options are sorted by maturity date, then strike price, then call before put.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = d (lowercase)
320	SecurityReqID	String [64]		Client generated ID supplied on the Security Definition Request.
322	SecurityResponseID	String [64]		Unique ID of a Security Definition message.
323	SecurityResponseType	int		Type of Security Definition message response. 1 = Accept security proposal as-is



Tag	Name	Data Type	Reqd	Comment
				<p>2 = Accept security proposal with revisions as indicated in the SecurityList (y) message</p> <p>5 = Reject security proposal. If the security (or an inverse of it) exists in the system, details will be provided in Symbol (55) and SecurityID (48). SecurityResponseType=Reject (323=5) is also returned if an inverse of the request security was created.</p>
58	Text	String [see comment]		<p>If the message reports a rejection (SecurityResponseType=Reject, 323=5) due to the security already being defined, the instrument data is provided in Symbol (55) and SecurityID (48).</p> <p>The maximum number of characters may exceed 128 for complex error messages.</p>
55	Symbol	String [255]		Common, human understood representation of the security.
48	SecurityID	String [10]		Security identifier value of SecurityIDSource (22) type.
22	SecuritySourceID	String		<p>Identifies the source of the SecurityID (48) value.</p> <p>Valid values:</p> <p>8 = Exchange defined. Available as Tradeable Instrument ID.</p>
	StandardTrailer		Y	

4.12.2. SecurityDefinition (d) Example – Accept As Is

```
8=FIXT.1.1|9=156|35=c|49=ABC01|56=ASX|34=331|52=20161123-05:13:08|320=ABC-c-20161123-05:13:08|321=1|167=MLEG|1378=0|555=2|600=IRH7|624=1|623=1|600=IRM8|624=2|623=2|55=[N/A]|10=238|
```



```
8=FIXT.1.1|9=0000156|35=d|49=ASX|56=ABC01|34=474|52=20161123-05:13:09.043|369=331|320=ABC-c-20161123-05:13:08|
322=RABC-c-20161123-05:13:08|323=1|55=UDC_IR-F_10MER|48=82157|22=8|10=069|
```

4.12.3. SecurityDefinition (d) Example – Accept As with Revisions

In this example the ratios were revised to the lowest common denominator.

```
8=FIXT.1.1|9=156|35=c|49=ABC01|56=ASX|34=340|52=20161123-05:15:22|320=ABC-c-20161123-05:15:22|321=1|167=MLEG|
1378=0|555=2|600=IRH7|624=1|623=4|600=IRM8|624=2|623=2|55=[N/A]|10=237|
```

```
8=FIXT.1.1|9=0000156|35=d|49=ASX|56=ABC01|34=484|52=20161123-05:15:22.175|369=340|320=ABC-c-20161123-05:15:22|
322=RABC-c-20161123-05:15:22|323=2|55=UDC_IR-F_13H57|48=82158|22=8|10=026|
```

4.12.4. SecurityDefinition (d) Example – Reject

In this example the definition was rejected because it already exists. Note the existing symbol is returned.

```
8=FIXT.1.1|9=156|35=c|49=ABC01|56=ASX|34=348|52=20161123-05:17:19|320=ABC-c-20161123-05:17:19|321=1|167=MLEG|
1378=0|555=2|600=IRH7|624=1|623=4|600=IRM8|624=2|623=2|55=[N/A]|10=005|
```

```
8=FIXT.1.1|9=0000290|35=d|49=ASX|56=ABC01|34=493|52=20161123-05:17:19.127|369=348|320=ABC-c-20161123-05:17:19|
322=RABC-c-20161123-05:17:19|323=5|55=UDC_IR-F_13H57|48=82158|22=8|58=Combination instrument UDC_IR-F_15ZDY
already exists or is an inverse of another combination. All is the same but buy/sell shifted|10=195|
```

4.13. OrderMassStatusRequest (AF)

The OrderMassStatusRequest message is used to request the status of active orders. It returns all orders where the requester was the last user to touch the order, unless a security has been specified to restrict the results. In addition, active orders for a specified Shared Order Group (SOG) can be returned when a SOG is provided in the request.

For each order, an ExecutionReport (8) message will be sent to notify the status of the order.

Note that there are specific cases when the OrderMassStatusRequest (AF) will not return open orders:

- Prior to the first pre-open state of an order book after a system start. The system is restarted every weekend.
- For existing SOG orders where the user is added to a SOG, until further activity occurs on the order, or the order is restored.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = AF
584	MassStatusReqID	String [64]	Y	Unique ID of mass status request as assigned by the institution.
585	MassStatusReqType	int	Y	Specifies the scope of the mass status request Valid values: 1 = Status for orders for a specified security 7 = Status for all active orders.



Tag	Name	Data Type	Reqd	Comment
1461	NoTargetPartyIDs	NumInGroup		Number of TargetPartyID (1462), TargetPartyIDSource (1463), and TargetPartyRole (1464) entries. It is only possible to specify one TargetParty; set to 1.
> 1462	TargetPartyID	String [see comment]	C	Used to identify source of TargetPartyID. Required if NoTargetPartyIDs > 0. Required if TargetPartyIDSource is specified. The maximum number of characters depends on the TargetPartyRole (1464) value. See TargetPartyRole (1464).
> 1463	TargetPartyIDSource	char	C	Identifies class or source of the TargetPartyID (1462) value. Required if TargetPartyID is specified. Valid values: D = Proprietary/Custom code.
> 1464	TargetPartyRole	int	C	The role of the party in the transaction. Required if TargetPartyID is specified. Valid values: 76 = Desk ID (Identifies the Shared Order Group (SOG) that this order is part of. If this is specified, all orders for the SOG will be reported, irrespective of the user that entered them. Up to 10 characters).



Tag	Name	Data Type	Reqd	Comment
55	Symbol	String [255]	Y	<p>Common, human understood representation of the security. Required if MassStatusReqType=Security (585=1) and SecurityID (48) are not specified.</p> <p>Required if SecurityID (48) is not present. If SecurityID (48) is specified, set Symbol (55) to [N/A].</p> <p>If MassStatusReqType=Security (585=1), message will be rejected if SecurityID (48) is present, and Symbol (55) is not [N/A].</p>
48	SecurityID	String [10]		<p>Security identifier value of SecurityIDSource (22) type. Requires SecurityIDSource (22).</p> <p>Symbol (55) must be set to [N/A] if SecurityID (48) is specified.</p> <p>If MassStatusReqType=Security (585=1), message will be rejected if SecurityID (48) is present, and Symbol (55) is not [N/A].</p>
22	SecurityIDSource	String		<p>Identifies the source of the SecurityID (48) value.</p> <p>Required if SecurityID (48) is specified.</p> <p>Valid values:</p> <p>8 = Exchange defined. Available as Tradeable Instrument ID.</p>
	StandardTrailer		Y	

4.13.1. OrderMassStatusRequest (AF) Example – No SOG Specified

Note orders with a SOG are present in the response because they were entered by this user. Other orders in the SOG are not returned.

```
8=FIXT.1.1|9=68|35=AF|49=ABCO2|56=ASX|34=56|52=20161124-04:30:52|584=ABC-AF-6|585=7|10=128|
```



```
8=FIXT.1.1|9=0000360|35=8|49=ASX|56=ABCO2|34=59|52=20161124-04:30:52.479|369=56|37=6207409765748850689|
198=6207409765748850689|11=ABCO1-2533|584=ABC-AF-6|911=3|912=N|453=5|448=ABC|447=D|452=1|448=ABCO2|447=D|452=12|
448=ABC|447=D|452=7|448=ASXCL|447=D|452=4|448=ABCO2|447=D|452=36|17=0|150=I|39=0|1=ABC1|581=1|55=IRZ9|48=65017|
22=8|54=1|38=2|40=2|44=96.5|15=AUD|59=0|151=2|14=0|6=0|10=211|
```

```
8=FIXT.1.1|9=0000382|35=8|49=ASX|56=ABCO2|34=60|52=20161124-04:30:52.479|369=56|37=6207409765815959553|
198=6207409765815959553|11=ABCO1-2534|584=ABC-AF-6|911=3|912=N|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=0|150=I|39=0|1=ABC1|
581=1|55=IRZ9|48=65017|22=8|54=1|38=3|40=2|44=96.5|15=AUD|59=0|151=3|14=0|6=0|10=216|
```

```
8=FIXT.1.1|9=0000382|35=8|49=ASX|56=ABCO2|34=61|52=20161124-04:30:52.479|369=56|37=6207409765845319681|
198=6207409765845319681|11=ABCO1-2535|584=ABC-AF-6|911=3|912=Y|453=6|448=ABC|447=D|452=1|448=ABCO4|447=D|452=12|
448=ABC|447=D|452=7|448=ASXCL|447=D|452=4|448=2246|447=D|452=76|448=ABCO2|447=D|452=36|17=0|150=I|39=0|1=ABC1|
581=1|55=APH7|48=58950|22=8|54=2|38=2|40=2|44=5431|15=AUD|59=0|151=2|14=0|6=0|10=197|
```

4.13.2. OrderMassStatusRequest (AF) Example – SOG Specified

Note all orders entered by this user are returned in addition to orders in this SOG entered by any SOG member

```
8=FIXT.1.1|9=100|35=AF|49=ABCO2|56=ASX|34=84|52=20161124-04:37:32|584=ABC-AF-7|585=7|1461=1|1462=2234|1463=D|
1464=76|10=085|
```

```
8=FIXT.1.1|9=0000384|35=8|49=ASX|56=ABCO2|34=109|52=20161124-04:37:32.945|369=84|37=6207411422020780033|
198=6207411422020780033|11=ABCO1-1755|584=ABC-AF-7|911=12|912=N|453=6|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=2234|447=D|452=76|448=ABCO1|447=D|452=36|17=0|150=I|39=0|1=ABC1|
581=1|55=IRZ9|48=65017|22=8|54=1|38=7|40=2|44=96.5|15=AUD|59=0|151=7|14=0|6=0|10=246|
```

... other execution reports not shown for brevity - note order with a SOG different to request and with no SOG ...

```
8=FIXT.1.1|9=0000362|35=8|49=ASX|56=ABCO2|34=117|52=20161124-04:37:32.946|369=84|37=6207409765748850689|
198=6207409765748850689|11=ABCO1-2533|584=ABC-AF-7|911=12|912=N|453=5|448=ABC|447=D|452=1|448=ABCO2|447=D|452=12|
448=ABC|447=D|452=7|448=ASXCL|447=D|452=4|448=ABCO2|447=D|452=36|17=0|150=I|39=0|1=ABC1|581=1|55=IRZ9|48=65017|
22=8|54=1|38=2|40=2|44=96.5|15=AUD|59=0|151=2|14=0|6=0|10=054|
```



```
8=FIXT.1.1|9=0000384|35=8|49=ASX|56=ABCO2|34=119|52=20161124-04:37:32.946|369=84|37=6207409765845319681|
198=6207409765845319681|11=ABCO1-2535|584=ABC-AF-7|911=12|912=N|453=6|448=ABC|447=D|452=1|448=ABCO4|447=D|452=12|
448=ABC|447=D|452=7|448=ASXCL|447=D|452=4|448=2246|447=D|452=76|448=ABCO2|447=D|452=36|17=0|150=I|39=0|1=ABC1|
581=1|55=APH7|48=58950|22=8|54=2|38=2|40=2|44=5431|15=AUD|59=0|151=2|14=0|6=0|10=038|
```

```
8=FIXT.1.1|9=0000384|35=8|49=ASX|56=ABCO2|34=120|52=20161124-04:37:32.946|369=84|37=6207411422175969281|
198=6207411422175969281|11=ABCO1-1757|584=ABC-AF-7|911=12|912=Y|453=6|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=2234|447=D|452=76|448=ABCO1|447=D|452=36|17=0|150=I|39=0|1=ABC1|
581=1|55=IRZ9|48=65017|22=8|54=1|38=9|40=2|44=96.5|15=AUD|59=0|151=9|14=0|6=0|10=051|
```

4.13.3. OrderMassStatusRequest (AF) Example – No Results

```
8=FIXT.1.1|9=69|35=AF|49=ABCO2|56=ASX|34=304|52=20161123-05:36:49|584=ABC-AF-3|585=7|10=182|
```

```
8=FIXT.1.1|9=0000141|35=8|49=ASX|56=ABCO2|34=330|52=20161123-05:36:49.712|369=304|37=0|584=ABC-AF-3|911=0|912=Y|
17=0|150=I|39=0|55=[N/A]|54=7|38=0|151=0|14=0|6=0|10=023|
```



4.14. OrderMassCancelRequest (q)

This message is used to cancel all orders for a FIX connection. A single Shared Order Group (SOG) can be specified to cancel all orders within the SOG in addition to orders for the FIX connection.

OrderMassCancelRequest can be used to cancel:

- All orders last touched by the user issuing the OrderMassCancelRequest (q)
- All orders in a SOG, including orders entered by other users
- All orders entered by another user. See OnBehalfOfCompID in the Standard Header.

In response to a mass cancel request, an OrderMassCancelReport (r) message will be sent to acknowledge receipt. MassCancelResponse (531) will indicate if the request can be processed using the value Cancel all orders (7), or rejected using the value *Cancel Request Rejected* (0).

For each cancelled order, an Execution Report will be sent with the OrdStatus showing the order as cancelled. For mass cancel requests cancelling orders of another user using OnBehalfOfCompID (115), only the owner of the orders will receive the Execution Report messages.

Note that messages for individual orders may be received before the OrderMassCancelReport (r) message.

If an OrderMassCancelRequest is sent at a time when one or more active orders are in a state that does not allow order cancellation, for example, the order book is in Maintenance, no action will be taken and no messages will be sent for those orders.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = q (lowercase)
11	ClOrdID	String [128]	Y	Unique identifier of this mass cancel request as assigned by the client.
530	MassCancelRequestType	char	Y	Specifies the type and scope of cancellation required. Valid values:



Tag	Name	Data Type	Reqd	Comment
				7 = Cancel all orders.
1461	NoTargetPartyIDs	NumInGroup		Number of TargetPartyID (1462), TargetPartyIDSource (1463), and TargetPartyRole (1464) entries. Only one allowed, set to 1.
> 1462	TargetPartyID	String [see comment]	C	Used to identify source of TargetPartyID. Required if NoTargetPartyIDs > 0. Required if TargetPartyIDSource is specified. The maximum number of characters depends on the TargetPartyRole (1464) value. See TargetPartyRole (1464).
> 1463	TargetPartyIDSource	char	C	Identifies class or source of the TargetPartyID (1462) value. Required if TargetPartyID is specified. Valid values: D = Proprietary/Custom code.
> 1464	TargetPartyRole	int	C	The role of the party in the transaction. Required if TargetPartyID is specified. Valid values: 76 = Desk ID. Identifies the shared order group (SOG) that the order is part of. If specified, all orders for that SOG will be cancelled, irrespective of the user that entered them. This will cancel orders in the SOG as well as any orders last touched by the user issuing the mass cancel request. Up to 10 characters.
60	TransactTime	UTCTimestamp	Y	Time this order request was initiated/released by the trader or trading system.



Tag	Name	Data Type	Reqd	Comment
	StandardTrailer		Y	

4.14.1. OrderMassCancelRequest (q) – No SOG

8=FIXT.1.1|9=103|35=q|49=ABCO1|56=ASX|34=416|52=20161123-05:29:01|11=ABC-q-20161123-05:29:01|530=7|60=20161123-05:29:01|10=100|

8=FIXT.1.1|9=0000399|35=8|49=ASX|56=ABCO1|34=589|52=20161123-05:29:01.825|369=415|37=6207042742833397761|198=6207042742833397761|11=ABC-G-20161123-04:35:19|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=6207042742833397761-1012|150=4|39=4|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=9|40=2|44=97.55|15=AUD|59=1|151=0|14=0|6=0|60=20161123-05:29:01.820|10=081|

8=FIXT.1.1|9=0000408|35=8|49=ASX|56=ABCO1|34=590|52=20161123-05:29:01.825|369=415|37=6207051244721160193|198=6207051244721160193|11=ABC-D-6|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=6207051244721160193-1013|150=4|39=4|1=ABC1|581=1|55=IRZ9|48=65017|22=8|54=1|38=2|40=2|44=97.6|15=AUD|59=6|126=20161124-04:41:24.000|151=0|14=0|6=0|60=20161123-05:29:01.820|10=145|

... other execution reports not shown for brevity ...

8=FIXT.1.1|9=0000383|35=8|49=ASX|56=ABCO1|34=601|52=20161123-05:29:01.827|369=416|37=6207060512463667201|198=6207060512463667201|11=ABCO1-32|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=6207060512463667201-1024|150=4|39=4|1=ABC1|581=1|55=APH7|48=58950|22=8|54=2|38=2|40=2|44=5431|15=AUD|59=0|151=0|14=0|6=0|60=20161123-05:29:01.820|10=192|

8=FIXT.1.1|9=0000118|35=r|49=ASX|56=ABCO1|34=602|52=20161123-05:29:01.831|369=416|11=ABC-q-20161123-05:29:01|1369=2016-11-23_1|530=7|531=7|10=160|

4.14.2. OrderMassCancelRequest (q) with SOG

8=FIXT.1.1|9=135|35=q|49=ABCO2|56=ASX|34=268|52=20161123-05:28:06|11=ABC-q-20161123-05:28:06|530=7|1461=1|1462=2234|1463=D|1464=76|60=20161123-05:28:06|10=086|



8=FIXT.1.1|9=0000405|35=8|49=ASX|56=ABCO2|34=285|52=20161123-05:28:06.076|369=267|37=6207060508181282817|
198=6207060508181282817|11=ABCO1-15|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=6207060508181282817-4399|150=4|39=4|1=ABC1|
581=1|55=IRZ9|48=65017|22=8|54=1|38=3|40=2|44=96.5|15=AUD|59=0|151=0|14=0|6=0|60=20161123-05:28:06.072|10=035|

... other execution reports not shown for brevity ...

8=FIXT.1.1|9=0000405|35=8|49=ASX|56=ABCO2|34=293|52=20161123-05:28:06.077|369=267|37=6207060512409141249|
198=6207060512409141249|11=ABCO1-31|453=6|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=2234|447=D|452=76|448=ABCO2|447=D|452=36|17=6207060512409141249-4407|150=4|39=4|1=ABC1|
581=1|55=APH7|48=58950|22=8|54=2|38=2|40=2|44=5431|15=AUD|59=0|151=0|14=0|6=0|60=20161123-05:28:06.072|10=225|

8=FIXT.1.1|9=0000150|35=r|49=ASX|56=ABCO2|34=294|52=20161123-05:28:06.077|369=268|11=ABC-q-20161123-05:28:06|
1369=2016-11-23_2|530=7|531=7|1461=1|1462=2234|1463=D|1464=76|10=143|



4.15. OrderMassCancelReport (r)

The OrderMassCancelReport (r) is an acknowledgement of the OrderMassCancelRequest. MassCancelResponse (531) will indicate if the request can be processed using the value Cancel all orders (7) or rejected using the value Cancel Request Rejected (0). Note that cancellation execution reports for individual orders may be received before the OrderMassCancelReport (r) message.

Tag	Name	Data Type	Reqd	Comment
	StandardHeader		Y	MsgType = r (lowercase)
11	ClOrdID	String [128]		Unique ID of this OrderMassCancelRequest as assigned by the institution.
1369	MassActionReportID	String [32]	Y	Unique ID for the order MassCancelReport assigned by the recipient of the OrderMassCancelRequest.
530	MassCancelRequestType	char	Y	Specifies the type and scope of cancellation required. Valid values: 7 = Cancel all orders. If OnBehalfOfCompID (115) is specified will cancel all orders for that user. If OnBehalfOfCompID not specified will cancel all orders for logged in FIX user.
531	MassCancelResponse	char	Y	Indicates the action taken by the counterparty order handling system as a result of the Cancel Request. Valid values: 0 = Cancel Request Rejected - See MassCancelRejectReason (532). 7 = Cancel All Orders.



Tag	Name	Data Type	Reqd	Comment
532	MassCancelRejectReason	int		Indicates why OrderMassCancelRequest was rejected. Required if MassCancelResponse = 0 Valid values: 0 = Mass cancel not supported 99 = Other.
1461	NoTargetPartyIDs	NumInGroup		Number of TargetPartyID (1462), TargetPartyIDSource (1463), and TargetPartyRole (1464) entries. It is only possible to specify one TargetParty; set to 1.
> 1462	TargetPartyID	String [see comment]		Used to identify source of TargetPartyID. Required if NoTargetPartyIDs > 0. Required if TargetPartyIDSource is specified. The maximum number of characters depends on the TargetPartyRole (1464) value. See TargetPartyRole (1464).
> 1463	TargetPartyIDSource	char		Identifies class or source of the TargetPartyID (1462) value. Required if TargetPartyID is specified. Valid values: D = Proprietary/Custom code.
> 1464	TargetPartyRole	int		The role of the party in the transaction. Required if TargetPartyID is specified. Valid values: 76 = Desk ID (Identifies the SOG that this order is part of. Up to 10 characters).
58	Text	String [see comment]		Will contain reject reason, if MassCancelResponse = 0.



Tag	Name	Data Type	Reqd	Comment
				The maximum number of characters may exceed 128 for complex error messages.
	StandardTrailer		Y	

4.15.1. OrderMassCancelReport (r)

8=FIXT.1.1|9=0000150|35=r|49=ASX|56=ABC02|34=294|52=20161123-05:28:06.077|369=268|11=ABC-q-20161123-05:28:06|1369=2016-11-23_2|530=7|531=7|1461=1|1462=2234|1463=D|1464=76|10=143|



4.16. 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
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.
> > 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).



Tag	Name	Data Type	Reqd	Comment
> > 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. The 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.
> > > 1767	RiskLimitAction	int		Risk protection action. Valid values: 4 = Warning.
> > > 1559	NoRiskWarningLevel	NumInGroup		Number of risk warning levels.



Tag	Name	Data Type	Reqd	Comment
> > > 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 this warning is being 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.
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.



4.16.1. PartyRiskLimitsReport (CM) Example

```
8=FIXT.1.1|9=0000307|35=CM|1128=9|49=ASX|56=ABCO9|34=72248|52=20161013-14:28:17.118|369=42800|  
1667=2809078_256_ABCO9_1889|1677=1|1671=1|1691=ABC1|1692=D|1693=24|1669=1|1529=1|1530=4|1767=4|1765=0.99|1559=1|  
1769=4|1560=0.90|1534=1|1535=1|1545=BB-S_SYMB|60=20161013-14:28:17.117|58=ABC1 exceeds 90.00%. Actual utilization  
99.50%.|10=114|
```



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	

5.1.1. News (B) Example

```
8=FIXT.1.1|9=0000146|35=B|49=ASX|56=ABC09|34=40836|52=20161123-05:33:03.817|369=21986|42=20161123-05:33:03.817|148=News|33=1|58=Volatility for : SFE-AP-O-2017M 16.760|10=166|
```



6. Message Sequencing

The following general information is provided to assist in understanding how messages may be sequenced between and within the ASX 24 FIX Order Entry gateways

6.1. Multiple gateways

ASX 24 offers a number of FIX Order Entry gateways for

- co-located participants; and
- remote participants accessing ASX 24 via ASX Net and ASX Net Global.

Each gateway operates independently of other gateways.

There is no guaranteed sequencing of messages across gateways. The observed variability in the performance of processing between gateways is such that the sequencing of messages is generally preserved across gateways.

Sequencing of messages may not be preserved between gateways under extreme load - characterised as messages arriving across multiple gateways with a time delta between messages that is significantly less than the typical latency across the gateway.

This effect may be observed both inbound (on the way to the matching engine) and, to a lesser extent, outbound (on the return from the matching engine).

6.2. Multiple sessions per gateway

Each gateway supports many simultaneous FIX sessions. Sessions are managed and processed in parallel.

There is no guaranteed sequencing between messages in different sessions on the same gateway. The observed variability in the performance of processing between sessions in a gateway is such that the sequencing of messages is generally preserved across sessions.



Sequencing of messages may not be preserved between sessions on a single gateway under extreme load - characterised as messages arriving across multiple sessions with a time delta that is significantly less than the typical latency across the gateway.

This effect may be observed both inbound (on the way to the matching engine) and, to a lesser extent, outbound (on the return from the matching engine).

6.3. Multiple instruments per session

Within a session, only one message per instrument may be in flight between the gateway and the matching engine.

The subsequent message in the same instrument will only be released by the gateway when the matching engine response to the message in flight has been received by the gateway. In the meantime, the gateway will release messages in other instruments in the same session.

There is therefore guaranteed sequencing of messages in the *same* instrument in a session but no guaranteed sequencing of messages in *different* instruments in the same session. Under typical load, the sequencing of messages in different instruments within a session is preserved.

Sequencing of messages may not be preserved if the gateway receives a message for an instrument which has a message in flight as the gateway will hold this message and will continue to release messages in other instruments that do not have a message in flight. This can occur when consecutive messages in an instrument and message(s) in other instrument(s) arrive with an offset that is less than the typical round trip latency between the gateway and the matching engine.

Sequencing of messages in the same instrument within a session is always guaranteed.

6.4. Matching engine

A single message sequence is established in the matching engine and preserved through the execution process.



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.

ASX®

