ASX Trade Refresh

ASX Trade Transactions

November 2024

Table of Contents

1 Introduction

Transactions are used to instruct ASX Trade to perform a certain action. Transactions are used to perform the following:

- Enter orders
- Amend and delete orders
- Enter, replace and delete quotes
- Send quote requests
- Enter trade reports
- Cancel trades
- Perform crossings
- Create Tailor Made Combinations (TMCs)
- Feed in prices for indices and other externally priced instruments
- Set Market Maker Protection parameters
- Indicate to ASX Trade that an OI application is ready to trade.

A transaction is validated by ASX Trade on entry, and if successful will be actioned by ASX Trade. Users will be informed of the status of their transaction (successful or rejected). Associated queries and broadcasts can then be used to track the progress/processing of the entered transaction.

For each transaction, the following information is provided in this document:

- Transaction Function - information about the transaction
- Transaction Properties - the facility required and the function call
- Message Structure - the structure of the message including a list of variables, their type and description.

1.1 Software Distribution Restrictions

Restrictions on the distribution of the OI software are detailed in the Developer's Agreement.

1.2 Supported Platforms

The following platforms are supported by ASX Trade:

- Linux Redhat Rhel6.10 x86 (32 and 64 bit)
- Linux Redhat Rhel7 x86 (32 and 64 bit)
- Windows 6.3 x86 (Windows Server 2012 R2 - 32 and 64 bit)
- Windows 10 x86 (Windows Server 2016 - 32 and 64 bit)

1.3 ASX Trade Support

For ASX Trade Open Interface Support, contact ASX Customer Technical Support (CTS) team either via email on cts@asx.com.au or phone 1800 663 053 (or on +61 2 9227 0372 from outside Australia).

1.4 ASX Trade OI Documentation Suite

ASX Trade Open Interface documentation has been created as a suite of documents that reference each other. The suite of documentation includes the following documents:

- ASX Trade Introduction and Business Information – This includes an introduction to ASX Trade for Open Interface developers and application providers. It also details business functionality to enable ASX Trade to be fully utilised.

- ASX Trade Open Interface Function Calls – This details the Open Interface function calls that enable communication between ASX Trade and the participant.
- ASX Trade Transactions – This contains the transactions that are used to instruct ASX Trade to perform particular actions.
- ASX Trade Queries – This details the queries that are used to retrieve information from ASX Trade.
- ASX Trade Broadcasts – This includes the broadcasts that are used to notify participants of an event or change occurring in ASX Trade.

1.5 Restrictions

Certain confidential information is prescribed by ASX as 'restricted information'. Details of what constitutes restricted information are set out below.

Some ASX Trade information is restricted information and may not be divulged to anyone who is not a Designated Trading Representative, except where that person is employed by an ASX trading participant and has a need to access that data as part of their duties.

1.5.1 Trading Participant Specific Information

Trading Participant Specific Information is the information specific to the trading participant that instigated a transaction on ASX Trade and which is not distributed by ASX to other participants. Participant Specific Information must not be divulged to anyone who is not a Designated Trading Representative of the trading participant, except where the person is employed by the trading participant and that person has a need to access that data as part of their duties.

Participant Specific Information includes, but is not limited to, the following:

- Client and Info references on orders and trades
- Total quantity for Iceberg orders and undisclosed quantities on orders
- The unique identifier of a trading participant allocated by ASX, i.e. the trading participant number, or the participant name in relation to products other than Listed Funds, Warrants and Structured products, Exchanged Traded Options and Futures.
- Some order types, e.g. shortsell
- Signum (user/session identifier) on orders and trades
- Expiry dates on orders
- Centre Point orders
- The short sell information on orders and trades
- Regulatory data
- Certain trade types e.g. BP (Booking Purpose); LN (Loan); LR (Loan Return)
- Booking reports that result from Unintentional Crossing Prevention.

Trading Participant Specific information is not included in enquiries where the order or trade does not belong to your trading participant ID.

1.5.2 Broker Service Providers

The trading participant may use dealing/information systems provided by an information vendor.

If your trading participant requests, ASX can provide the vendor with:

- All the trading participant's specific information as detailed in *Trading Participant Specific Information* above.

- The vendor can then integrate this information into their dealing/information systems for the trading participant.
- A vendor that has access to Trading Participant Specific information is known as a Broker Service Provider (BSP).
- The BSP must keep this Trading Participant Specific Information confidential and must not collate or distribute this information to anyone other than the relevant trading participant.

1.6 Version History

This document has been revised according to the table below:

Version	Date	Comment
v1.1	February 2015	• MO1 – Changes made to exchange_info_t (named structure 50004) to clarify that this structure is overlaid with asx_exchange_info_t for Sweep orders. • MO3 – Changes made to exchange_info_t (named structure 50004) to clarify that this structure is overlaid with asx_exchange_info_t for Sweep orders.
v2.0	March 2015	The following has been added: • MO1 - New optional sub-structure enhanced_cp_matching_t (named structure 34831) is available for the MO1 Single Order Update transaction. This sub-structure is reserved for future use with Centre Point and Sweep orders. • MO1 - Changes added to centre_point_order_t (named structure 34816). • MO3 – New optional sub structure enhanced_cp_matching_t (named structure 34831) has been added. This is reserved for future use with Centre Point and Sweep orders. • MO3 – Changes made to centre_point_order_t (named structure 34816).
v2.1	October 2018	• Updated to new ASX branding • Removal of market and instrument group, which are now covered in https://www.asxonline.com/content/dam/asxonline/public/documents/asx-trade-refresh-manuals/asx-trade-markets-instrument-groups-and-trade-condition-codes.pdf
v3.0	September 2019	• Updated for ASX Trade Refresh – MO31, MO33, MO98, MO99 removed – All references to activation of inactive orders removed
v3.1	October 2019	• Updates for ASX Trade Refresh • MO1 centre_point_order_t > mid_tick_c update • MO1 Transaction Properties Correction • Clearing_info_t > open_close_req_c clarification
V3.2	November 2019	• Operating System Support for Rhel6.10 x86 (32 and 64 bit) • Section 3.3 (CC19) update to instance_c • New Section (CC86) 4.4 Return Codes • Update to exchange_order_type_n – removal of 128 = inactive in section 11.3.3 • Update to description for short_sell_quantity_i in sections 11.3.12 and 12.3.11 • Update to description of Changes to Quantity in Section 12.1.2 • Update to settlement_date_s – if set to NULL or spaces this will result in a system calculated settlement date in sections 18.3, 19.3.1, 20.3.2

Version	Date	Comment
		• New Sections (MO76) – 21.3.8 and 22.3.9 Return Codes
v3.3	February 2020	• Replacing MO36 quote with FoK/FaK description in section 14.1 • Return code note added under MO36 in section 14.3.3
v3.4	April 2020	• Updated exch_order_type_n variable description in section 12.3.3 under basic_order_update_t structure
v3.5	August 2020	• Updated short_sell_quantity_i variable description in section 12.3.11 under short_sell_order_t structure
v3.6	April 2021	• Updated section 12.3.12 enhanced_cp_matching_t (named structure 34831)
V3.7	June 2022	• Updated series_t description in Section 4.3 under cl_enter_delayed_trade_report_t • Updated series_t description in Section 5.3 under cl_cancel_initial_trade_report_t
V3.8	November 2024	• Updated Section 12.1 MO3 Single Order Update – cross protocol order amend not permitted

2 Common Structures

There are common structures that can be found in the majority of messages. This includes unique identifiers and series structure.

2.1 Unique Identifiers - transaction_type_t and broadcast_type_t

Every message has a unique identifier making it possible for users to interpret the content. The identifier is made up of two letters and a number. For transactions and queries, the structure that holds these identifying values is the transaction_type_t. For broadcasts, it is the broadcast_type_t. Both structures are identical, and are displayed in the table below.

Variable	Description
central_module_c	char[1] The Central Module defines which subsystem handles or issues the message. Some samples of the letters indicating the central modules are: M = Matching Engine (ME) C = Clearing (CL) I = Information (IN) D = Common Database (CDB) O = Operation (OP) L = List Module (LM) U = Supervision (SU).
server_type_c	char[1] The Server Type describes the type of the operation that the message will generate. Some samples of the letters indicating the server types are: O = Order Q = Query A = Answer D = Deal C = Command I = Information B = Broadcast.
transaction_number_n	uint16_t The transaction number is a numerical value used to distinguish between different message types.

2.2 Series Structure – series_t

The series_t structure appears in most messages to identify the products being traded, queried or broadcasted.

Depending on the message, there are different requirements for which series_t sub fields contain data, and which are filled with binary zeros. These requirements are documented in each message structure as required.

Variable	Description
country_c	uint8_t Country and/or exchange identity. For ASX, the value here should be set to 15 and the number can be considered as constant.

Variable	Description
market_c	uint8_t An integer representing the market code. Zero can be used to act as a filter or wildcard. For an entire list of possible values, see *Appendix 1 – Markets.*
instrument_group_c	uint8_t A numerical value indicating the instrument group. Zero can be used to act as a filter or wildcard. For an entire list of possible values see *Appendix 2 - Instrument Groups*.
modifier_c	uint8_t Expiration date modifier. This value is set to zero when the instrument is new. The value is incremented by one each time the instrument is involved in an issue, split, etc. Note that the modifier value can be different for bid and ask options in the same series. The modifier can also be used to indicate a special market. In this case the modifier will be >= 200. Refer to *ASX Trade Introduction and Business Information* for more information.
commodity_n	uint16_t A numerical value indicating the commodity (underlying). Example values are: 20046 = ASX 5080 = BHP.
expiration_date_n	uint16_t Expiration date of the financial instrument. Note this only applies to derivatives. Equities will have zero in this field. A bit pattern is used. The seven most significant bits are used for year, the next four for month, and the five least significant bits for day. All these bits make up an unsigned word. The year-field starts counting from 1990. Thus, 1990 = 1, 1991=2 ... 2001=12. E.g. January 1, 1990 would be represented in binary as: 0000001 0001 00001, and in decimal: 545.
strike_price_i	int32_t The strike price is a part of the binary series for derivatives. Equities will have zero in this field. This is always an integer. The implicit number of decimals to be used can be determined by a field that is associated with each instrument class. Refer to *DQ122 Query Delta Instrument Class* in *ASX Trade Queries* for more information.

3 CC19 Cancel Trade

3.1 Transaction Function

This transaction is used for cancelling the participant's side of a trade. The trade is not actually cancelled until both parties involved in the trade enter this transaction.

The ext_seq_nbr_i field must be specified in the message to identify the trade.

3.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP7
Struct Name	cancel_trade_t
Partitioned	false

3.3 Message Structure

3.3.1 cancel_trade_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'C', 19}.
series	series_t The series for which the trade exists. The series must have a value assigned to every field; it does **not** act as a wild card whereby some fields can be ignored.
instance_c	uint8_t Set to one.
bought_or_sold_c	uint8_t The side of the trade that is being cancelled. Possible values include: 1 = Bid 2 = Ask.
filler_2_s	char[2] Ignore. Used for byte alignment.
trade_number_i	int32_t Ignore. Currently not used.
ext_seq_nbr_i	int32_t ASX Trade Slip number. Of the format 1OPNNNNNNN 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC NNNNNNN - Trade Number.

4 CC86 Delayed Trade Report Entry

4.1 Transaction Function

This transaction is used for registering a Delayed Trade Report that was previously reported as an Initial Trade Report. For more information on trade reporting refer to *Trade Reporting* in *ASX Trade Introduction and Business Information*.

The transaction includes a series, instance and a trade slip number, uniquely identifying the Initial Trade Report. These fields must be set to the corresponding values of the Initial Trade Report.

4.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP7
Struct Name	cl_enter_delayed_trade_report_t
Partitioned	false

4.3 Message Structure

4.3.1 cl_enter_delayed_trade_report_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'C', 86}.
series	series_t The series for which the initial trade report exists. The series must have a value assigned to every field however the validation only requires that instrument type matches the trade being cancelled.
instance_c	uint8_t Set to zero.
filler_3_s	char[3] Ignore. Used for byte alignment.
ext_seq_nbr_i	int32_t ASX Trade Slip number. Of the format 1OPNNNNNNN: 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC NNNNNNN - Trade Number.

4.3.2 Return Codes

Cstatus	Txstat	Description
Successful	700001	Transaction Successful
Invalid Series	-700132	Invalid series in transaction.

5 CC87 Cancel Initial Trade Report

5.1 Transaction Function

This transaction is used for cancelling an Initial Trade Report that was previously entered. For more information on trade reporting refer to *Trade Reporting* in *ASX Trade Introduction and Business Information*.

The transaction includes a series, instance and a trade slip number, uniquely identifying the Initial Trade Report. These fields must be set to the corresponding values of the Initial Trade Report.

5.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP7
Struct Name	cl_cancel_initial_trade_report_t
Partitioned	false

5.3 Message Structure

5.3.1 cl_cancel_initial_trade_report_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'C', 87}.
series	series_t The series for which the trade report exists. The series must have a value assigned to every field however the validation only requires that instrument type matches the trade being cancelled.
instance_c	uint8_t Set to zero.
filler_3_s	char[3] Ignore. Used for byte alignment.
ext_seq_nbr_i	int32_t ASX Trade Slip number. Of the format 1OPNNNNNNN: 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC NNNNNNN - Trade Number.

6 DC3 Add Tailor Made Combination

6.1 Transaction Function

This transaction is used to add a Tailor Made Combination (TMC). The transaction is sent as a query, as the added combination series identifier is returned as an answer.

Note:
The individual series legs in a combination cannot go across several Matching Engine (ME) partitions. All legs in the combination must exist in the same ME partition.

6.2 Query Properties

Function Call	omniapi_query_ex
Facility	EP5
Struct Name	add_tm_combo_t
Partitioned	false
Segmented	false
Answers	DI3

6.3 Answer Properties

Transaction Type	DI3
Struct Name	answer_add_tm_combo_t
Segmented	false

6.4 Message Structure

6.4.1 add_tm_combo_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'C', 3}.
series	series_t Set to zeros.
no_of_legs_n	uint16_t Number of legs in the combination.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	add_tm_combo_item_t[4] Each item contains one leg of the combination. See add_tm_combo_item_t sub structure below.

6.4.2 add_tm_combo_item_t

Variable	Description
series	series_t The series for this particular leg of the combination. The series must have a value assigned to every field; it does **not** act as a wildcard whereby some fields can be ignored.
ratio_n	uint16_t Relative numbers of contracts. This ratio must be reduced to the lowest common denominator as per the example of 1:2 given below. The transaction would fail if users tried to create a combination with ratio 10:20. Consult the rules of *TMCs in TMC Programming Guidelines* in *ASX Trade Introduction and Business Information*.
op_if_buy_c	char[1] Specifies whether to buy or sell the series when buying the combination. Possible values: B = Buy S = Sell.
op_if_sell_c	char[1] Specifies whether to buy or sell the series when selling the combination. Possible values: B = Buy S = Sell.

6.4.2.1 Examples

This input creates a combination where instrument 1 is bought and instrument 2 is sold to a ratio 1:2 when buying the combination.

Variable	Value
no_of_legs_n	2
Instrument 1	
ration_n	1
op_if_buy_c	B
op_if_sell_c	S
Instrument 2	
ratio_n	2
op_if_buy_c	S
op_if_sell_c	B

6.5 Answer Structure

The answer contains the binary code of the series that identifies the newly created TMC. This is disseminated by the BU126 broadcast (refer to *BU126 Combination Series Update* in *ASX Trade Broadcasts)*.

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'I', 3}.
series	series_t The series for the combination.

7 DC87 Set Market Maker Protection

7.1 Transaction Function

This transaction is used by Market Makers to set and change Market Maker Protection parameters for an underlying.

7.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	set_mm_protection_t
Partitioned	false

7.3 Message Structure

7.3.1 set_mm_protection_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'C', 87}.
series	series_t Not used in this transaction. Set to zeros.
da87	da87_t See da87_t sub structure below.

7.3.2 da87_t

Variable	Description
quantity_protection_q	int64_t Specifies the limit of the total traded contracts per underlying within the exposure time interval when Market Maker Protection is triggered. When this value is reached or exceeded the system automatically removes all quotes for the instruments connected to the underlying. A value of zero means that no Quantity Protection exists.
delta_protection_q	int64_t Specifies the limit of the delta value per underlying within the exposure time interval when Market Maker Protection is triggered. When this value is reached or exceeded the system automatically removes all quotes for the instruments connected to the underlying. A value of zero means that no Delta Protection exists.
exposure_time_interval_i	int32_t Specifies the rolling time interval in milliseconds used in Quantity/Delta Protection calculations. The provided value when used by the system will be rounded up to the nearest multiple of 1,000; therefore the practical minimum value is 1,000ms. A value of zero means that Market Maker Protection functionality is turned off for the applicable underlying.

Variable	Description
frozen_time_i	int32_t Specifies the time interval in milliseconds when quotes are rejected after Market Maker Protection has been triggered. A value of zero means that quotes are considered as frozen for the rest of the day.
commodity_n	uint16_t The code for the underlying that the Market Maker Protection parameters in this item apply to.
country_id_s	char[2] Not used for this transaction.
ex_customer_s	char[5] Not used for this transaction.
include_futures_c	uint8_t Specifies if Futures and Forwards are to be included in the Delta Protection calculation. The group_type_c of the instrument group definition determines whether the instrument is a Future or Forward (i.e. group_type_c equals 2 or 3). Possible values: 1 = Yes, include Futures and Forwards in the Delta Protection calculation 2 = No, do not include Futures and Forwards in the Delta Protection calculation.
filler_2_s	char[2] Ignore. Used for byte alignment.

8 II2049 Index Underlying Update

8.1 Transaction Function

This transaction is used for feeding index and other externally priced instrument prices into the market. This transaction is a restricted transaction and is only available to specific users.

The message can be used to update the following data for indices:

- High
- Low
- Last
- Points change since opening
- Percentage change since opening
- Percentage change since last
- Closing Price
- Time of update (ext_time_s).

The message can be used to update the following data for other externally priced instruments:

- Open
- High
- Low
- Last
- Time of update (ext_time_s).

Data entered through this transaction is disseminated to the market using the *BD2 Edited Price Information* broadcast, as detailed in *ASX Trade Broadcasts*.

8.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	underlying_info_update_t
Partitioned	false

8.3 Message Structure

8.3.1 underlying_info_update_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'I', 'I', 2049}.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	underlying_info_update_item_t[500]

Variable	Description
	The array of information for each underlying or index. See underlying_info_update_item_t sub structure below.

8.3.2 underlying_info_update_item_t

Variable	Description
series	series_t The index or other externally priced series for which values are being entered.
bid_premium_i	int32_t Ignore. Currently not used.
ask_premium_i	int32_t Ignore. Currently not used.
closing_price_i	int32_t Use this field to enter the previous day's closing price of the series.
opening_price_i	int32_t Price of series at the opening of on-market trading on the current trading day. Not used for index.
high_price_i	int32_t Highest price of series or value of index attained during the day.
low_price_i	int32_t Lowest price of series or value of index attained during the day.
last_price_i	int32_t Last price of series or last value of index.
ref_price_i	int32_t Ignore. Currently not used.
turnover_u	int64_t Ignore. Currently not used.
best_bid_volume_u	int64_t Ignore. Currently not used.
best_ask_volume_u	int64_t Ignore. Currently not used.
undisclosed_bid_volume_c	uint8_t Ignore. Set to 2.
undisclosed_ask_volume_c	uint8_t Ignore. Set to 2.
isin_code_s	char[12] Ignore. Currently not used. Fill with spaces.
ext_time_s	char[6] The time of the update in 'HHMMSS' format.
change_previous_i	int32_t Percentage change since the last disseminated value for indices. The value contains the same number of decimal places as defined in the index series. Not used for other externally priced series.
change_yesterday_i	int32_t

Variable	Description
	Percentage change since current day's initial value for indices. The value contains the same number of decimal places as defined in the index series. Not used for other externally priced series.
points_of_movement_i	int32_t Points change since the current day's initial value for indices. The value contains the same number of decimal places as defined in the index series. Not used for other externally priced series.

9 MC2 Crossing Quote Request

9.1 Transaction Function

This transaction is used for requesting a crossing market quote for derivatives crossings (Cross with Book or Two Sided Crossing).

Users that want to perform a derivatives crossing must first send in this Crossing Quote Request transaction for the total volume to be crossed, as a bid/ask request. This has to be done even if the market is established.

When the transaction is successful, ASX Trade will issue a Quote Request broadcast (MI4) to the Market Makers who have obligations in the class to which the particular series belongs. Where there are no Market Makers, the quote request broadcast is sent to all trading participants. If the transaction is not successful, no broadcast will be sent.

Using this transaction is just the first step to undertake a derivatives crossing. The full procedure is described in *Derivatives Crossings* in *ASX Trade Introduction and Business Information*.

9.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	crossing_request_t
Partitioned	true

9.3 Message Structure

9.3.1 crossing_request_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'C', 2}.
series	series_t The series for which the crossing quote request is made. The series must have a value assigned to every field; it does **not** act as a wildcard whereby some fields can be ignored.
mp_quantity_i	int64_t The total quantity to be crossed must be specified here.
block_n	uint32_t Block Size - always set to one.
bid_or_ask_c	uint8_t Bid or Ask. Although the OI allows the value of one and two to indicate the user requires a bid or ask quote respectively, users should set this value to zero. The resultant MI4 broadcast will indicate to the Market Maker that both a bid and ask quote is required.
filler_3_s	char[3] Ignore. Used for byte alignment.

9.3.2 Return Codes

An MC2 transaction may be aborted by ASX Trade. A reason for the error is provided in the transaction status parameter.

Cstatus	Txstat	Description
Successful	0	Request accepted.
Transaction aborted	LM_MMSUP_INV_ORDERTYPE	Invalid order type, must be either bid or ask.
Transaction aborted	LM_MMSUP_ILL_TRT_IN_USR	Not allowed to place an order, quote request not legitimate.
Transaction aborted	LM_MMSUP_COMM_STOP	Series has been stopped for trading.
Transaction aborted	LM_CROSS_NOT_USER	User is not found in user database.
Transaction aborted	LM_CROSS_INSID_NOTFND	The instrument ID (series + block size) is not found.
Transaction aborted	LM_CROSS_INT_NOT_ALLOWED	Crossing market is not allowed for this instrument type.
Transaction aborted	LM_CROSS_REQ_MEM_ACT	Request denied, member has a Crossing Market active.
Transaction aborted	LM_CROSS_REQ_BLOCKED	Request denied, instrument blocked by a previous Crossing Market.

10 MC4 Quote Request with Volume

10.1 Transaction Function

This transaction allows participants to request quotes. The quote request goes to ASX Trade and if there is no price, the request will be broadcasted. Obligated Market Makers receive this Quote Request broadcast in their nominated underlying stocks. If there are no obligated Market Makers then the broadcast will go to all participants.

Refer to the ASX website for minimum quantities, spread and tick size limits. This information is located at: http://www.asx.com.au/products/equity-options/market-making.htm.

10.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	quote_request_vol_t
Partitioned	true

10.3 Message Structure

10.3.1 quote_request_vol_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'C', 4}.
series	series_t The series for which the quote request is sent. The series must have a value assigned to every field; it does **not** act as a wildcard whereby some fields can be ignored.
block_n	uint32_t Block Size - always set to one.
bid_or_ask_c	uint8_t Bid or Ask. Values include: 0 = Bid and Ask, user wants both a bid and ask quote sent 1 = Bid, user wants a bid quote sent 2 = Ask, user wants an ask quote to be sent.
filler_3_s	char[3] Ignore. Used for byte alignment.
mp_quantity_i	int64_t Quantity required. A value of zero indicates a quote with any volume is required.

10.3.2 Return Codes

After a successful MC4 transaction, the quote request is sent to connected applications through the MI4 broadcast.

Cstatus	Txstat	Description
Transaction aborted	LM_MMSUP_NOT	Quote request not legitimate. Price exists in given series.

11 MO1 Single Order Insert

11.1 Transaction Function

The MO1 transaction is used for entering orders. It returns an order identifier that can be used to track the order. Users wishing to keep track of their own orders need to subscribe to the BO5 Firm Order Book broadcast (refer to *BO5 Firm Order Book* in *ASX Trade Broadcasts*).

The transaction is restricted by the Transactions per Second (TPS) rate set by ASX.

The following order types are supported by MO1:

- Limit (LMT)
- Market-to-Limit (MTL)
- Best-Limit (BST)
- Market Bid (TKO) (only ASX Trading Operations can enter this)
- Price Stabilisation (PST) - only certain users can enter these, as allowed by ASX Trading Operations
- Short Sell
- Undisclosed Quantity
- Centre Point Limit and Centre Point Market
- Centre Point Block Limit and Centre Point Block Market
- Limit Sweep and Market-to-Limit Sweep
- Imbalance Limit.

In conjunction with the order types there are also several validity attributes:

- FoK - Fill or Kill, cancelled if all the quantity cannot be executed immediately.
- FaK - Fill and Kill, fill the order now as far as possible then cancel the rest.
- Day - Expire at the end of the day.
- Exp - Expire at the instrument's expiration date.
- Date - Expire after a certain amount of calendar days.
- GTC - Good till Cancel, expire at the maximum allowed time for that particular instrument type.

11.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	The transaction complies with the VIT concept. The top most struct is single_order_insert_t (named structure 34808).
Partitioned	true

The MO1 is a variable information transaction. Sub-headers within the message identify what is contained in the message. The overall structure is:

- single_order_insert_t (named structure 34808)
- one or more sequences of:
  - sub_item_hdr_t
  - a choice of:

- basic_order_t (named structure 34810)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- regulatory_t (named structure 34821)
- reserve_order_t (named structure 34812)
- centre_point_order_t (named structure 34816)
- enhanced_cp_matching_t (named structure 34831)
- crossing_t (named structure 34820)
- inactive_order_t (named structure 34818)
- short_sell_order_t (named structure 34829).

The transaction must always have the series and items_n populated. The order entered is defined by having one or several sub structures appended to the main structure. Sub structures can be defined in any order and only required sub structures should be used. If a sub structure is not provided, default or blank values will be assumed. size_n is not used for this transaction.

As an example, the sequence of sub structures for entering a Limit order is as follows:

- basic_order_t (named structure 34810)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- regulatory_t (named structure 34821)

Sub structure basic_order_t is mandatory. The other sub structures are optional and should only be included, if required.

The sequence of sub structures for entering Centre Point and Centre Point Block orders is as follows:

- basic_order_t (named structure 34810)
- centre_point_order_t (named structure 34816)
- enhanced_cp_matching_t (named structure 34831)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- regulatory_t (named structure 34821).

The sequence of sub structures for entering a Limit Sweep Iceberg order is as follows:

- basic_order_t (named structure 34810)
- centre_point_order_t (named structure 34816)
- reserve_order_t (named structure 34812)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- regulatory_t (named structure 34821).

The sequence of sub structures for entering a Short Sell order is as follows:

- basic_order_t (named structure 34810)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- regulatory_t (named structure 34821)
- short_sell_order_t (named structure 34829).

11.3 Message Structure

11.3.1 single_order_insert_t (named structure 34808)

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 1}.
series	series_t The series for which the order exists. The series must have a value assigned to every field; it does NOT act as a wildcard whereby some fields can be ignored.
items_n	uint16_t The number of sub-items following this top structure.
size_n	uint16_t Not used for this transaction.

11.3.2 sub_item_hdr_t

Variable	Description
named_struct_n	uint16_t Contains a number, indicating the type of structure that follows.
size_n	uint16_t Not used for this transaction.

11.3.3 basic_order_t (named structure 34810)

Variable	Description
premium_i	int32_t The price of the order. A combination of this field and the order_type_c field signify different types of orders. 0 = Market order. Any other value than zero = Limit order (order_type_c = 1 or 65). The price for a TMC order can be positive, zero or negative.
quantity_i	int64_t Quantity of the order. For iceberg orders this the total quantity of the order. The shown quantity portion of the iceberg order is defined in the reserve_order_t (named structure 34812) sub structure.
block_n	uint32_t Block size. Possible values: 0 = Fill or Kill order (time_validity_n = 0) 1 = All other types of orders.

Variable	Description
time_validity_n	uint16_t This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte). Unit = 0, Count = 0 (i.e. binary = 0): Order is an “Immediate” type order. Fill Or Kill when block_n = 0 or Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not have an expiry date, if this order is for an equity it will be valid for the maximum allowed time for that particular instrument type. Unit = 5, Count = a positive integer (i.e. binary 101 0000 0011, hex 503, dec 1283): Order will be valid for that many calendar days, including today. Expiry will occur at the end of day’s trading on the final day. The answer contains the number of days left for the order, decreasing by one every day. It does NOT contain the number of days when the order was originally placed. Unit = 6, Count = 0 (i.e. binary 110 0000 0000, hex 600, dec 1536): Order is “Good Till Cancelled” type. Order will be valid for the maximum allowed time for that particular instrument type. Centre Point and Centre Point Block orders are not carried over to the next day, i.e. they are purged at the end of the day, regardless of the entered time_validity_n. For Centre Point Block orders, validity FoK is only allowed when the MAQ (Minimum Acceptable Quantity) is zero or equal to the order quantity.
exch_order_type_n	uint16_t Exchange specific order types. 2 = Short Sell order *(premium_i = an integer and order_type_c = 1: Limit order* *premium_i = 0 and order_type_c = 2: Market order* *premium_i = 0 and order_type_c = 3: Market-to-Limit order* *premium_i = 0, order_type_c = 17 and time_validity_n != 0: Best-Limit order).* Can also be combined with other exchange specific order types outlined below. 4 = Market Bid order *(premium_i = an integer and order_type_c = 1, only entered by ASX Trading Operations)* 8 = Price Stabilisation/Green Shoe order *(premium_i = an integer, order_type_c = 1 and time_validity_n !=0)* 32 = Undisclosed order (use order_type_c to determine order type). 64 = Centre Point Order (use order_type_c to determine Market or Limit). 2048 = Sweep order (use order_type_c to determine Market-to-Limit or Limit) 4096 = Centre Point Block order (use order_type_c to determine Market or Limit).
order_type_c	uint8_t Order type – a combination of this field and the premium_i field signifies different types of orders. Possible values: 1 = Limit price order (premium_i = an integer) 2 = Market order (premium_i = 0) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0 and time_validity_n !=0) 65 = Imbalance Limit order (premium_i = an integer). Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Sweep, Centre Point and Centre Point Block orders.

Variable	Description
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
filler_2_s	char[2] Ignore. Used for byte alignment.

11.3.4 reserve_order_t (named structure 34812)

Variable	Description
display_quantity_i	int64_t The shown quantity of an iceberg order. The reserve_order_t sub-structure only needs to be specified when entering iceberg orders. Can also be used for Limit Sweep order types to enter them as an iceberg order.
original_display_quantity_i	int64_t Not used in the MO1 and MO3 transactions. Set to zero.

11.3.5 centre_point_order_t (named structure 34816)

Variable	Description
minimum_quantity_i	int64_t MAQ of Centre Point Block order, Any Price Block order or Limit Sweep order when executing in Centre Point. Specifies the minimum quantity that must be traded in each execution cycle. 0 = no minimum acceptable quantity. Must be set to zero for Centre Point orders (exch_order_type_n = 64 or 66).
mid_tick_c	uint8_t Specifies whether the limit price of a Centre Point Limit or Centre Point Block Limit order should be a half-tick more aggressive (i.e. improved) and/or allowed for permitted prices other than mid-point (‘dark limit’ order). Specifies whether a Limit Sweep order is eligible for passive execution in Centre Point at a half-tick above the limit price. 1 = mid-tick attribute set on 2 = mid-tick attribute set off 3 = allowed for permitted prices in addition to mid-point (‘dark limit’ order) 4 = allowed for permitted prices in addition to mid-point (‘dark limit’ order), with mid-tick attribute set on 5 = Any Price Block order (can only be used with Centre Point Block Limit orders) 6 = Any Price Block order with mid-tick attribute set on (can only be used with Centre Point Block Limit orders). Limit Sweep orders fully integrate the liquidity in Centre Point and ASX TradeMatch and will interact with both ‘mid-point only’ and ‘dark limit’ Centre Point and Centre Point Block orders and Any Price Block orders. The only allowed mid_tick_c values for Limit Sweep orders are 1 and 2.
preference_only_c	uint8_t Specifies whether a Centre Point order or a Centre Point Block order is a Preference and Kill order or not. For Limit Sweep orders, only ‘0’ or ‘2’ are valid values.

Variable	Description
	0 = not defined or 'no', order is not a Preference and Kill order. 1 = 'yes', order is a Preference and Kill order. Time validity must be set to Fill and Kill or Fill or Kill for this option (this is currently not supported). 2 = 'no', order is not a Preference and Kill order.
single_fill_minimum_quantity_c	uint8_t Specifies whether the minimum acceptable quantity (minimum_quantity_i) of Centre Point Block or Limit Sweep orders must be satisfied in a single fill or not. 0 = not defined 1 = MAQ must be satisfied in a single fill 2 = MAQ may be satisfied in multiple fills (aggregated execution). Must be set to 0 or 2 for Centre Point orders (exch_order_type_n = 64 or 66). Can only be set to 1 for Centre Point Block and Limit Sweep orders that have a minimum_quantity_i > 0.
filler_1_s	char[1] Ignore. Used for byte alignment.

11.3.6 inactive_order_t (named structure 34818)

Variable	Description
inactive_c	uint8_t Specifies whether an order should be entered as a central inactive order. 0 = not defined 2 = active order.
filler_3_s	char[3] Ignore. Used for byte alignment.

11.3.7 exchange_info_t (named structure 50004)

Variable	Description
exchange_info_s	char[32] A free text field used at the participant's discretion. **Note**: When entering Centre Point or Sweep orders it is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.

11.3.7.1 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.
boq_list_s	char[6] Ignore. Not used when entering orders with MO1.
initial_trd_report_c	uint8_t Ignore. Not used when entering orders with MO1.
filler_1_s	char[1] Ignore. Used for byte alignment.

Variable	Description
extended_price_q	int64_i This field may be set to a trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000) for Centre Point orders trading at an extended price, or the special value indicating that it is to be ignored. If the 63rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

11.3.8 free_text_t (named structure 34801)

Variable	Description
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.
filler_1_s	char[1] Ignore. Used for byte alignment.

11.3.9 clearing_info_t (named structure 34802)

Variable	Description
give_up_member	give_up_member_t The clearing identifier used for the order. See give_up_member_t below.
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
open_close_req_c	uint8_t Set to Zero
filler_1_s	char[1] Ignore. Used for byte alignment.

11.3.9.1 give_up_member_t (named structure 50002)

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to 'AU', indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the Clearing Participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

11.3.10 crossing_t (named structure 34820)

Variable	Description
crossing_key_i	int32_t

Variable	Description
	Crossing key for Unintentional Crossing Prevention. When two orders from the same participant with the same crossing key trade out, the resulting trade is treated like a booked transaction and not published to the market as a trade. Setting this field to zero for an order means “no Unintentional Crossing Prevention” for this order.

11.3.11 regulatory_t (named structure 34821)

Variable	Description
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX Specific Overlay of regulatory_data_s Variable below.

11.3.11.1 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N = False
execution_venue_s	char[4]	2 to 5	Execution venue. Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

11.3.12 short_sell_order_t (named structure 34829)

Variable	Description
short_sell_quantity_i	int64_t Partial short quantity of a short sell order. For orders that are not short sell orders (2nd least significant bit not set), this sub-structure should not be included at all. For orders that are short sell orders (2nd least significant bit is set), must be equal to or less than the total order quantity and greater than zero.

11.3.13 enhanced_cp_matching_t (named structure 34831)

This optional sub-structure is reserved for future use with Centre Point and Sweep orders. Functionality relating to these fields will be detailed at a later stage. Until then, this sub-structure should not be used.

Variable	Description
participant_order_attribute_i	uint32_t Currently not supported
order_attributes_i	uint32_t Currently not supported

11.3.14 Return Codes

After a successful MO1 transaction, an order number and information regarding the state of the order will be returned. For a standard combination order, each leg will get the same order number.

Cstatus	Txstat	ordidt
Successful	1 - No part of the order placed in the order book and no part closed (Fill and Kill only).	order number
Successful	2 - Whole order closed.	order number
Successful	3 - Order partially closed and nothing placed in order book.	order number
Successful	4 - Whole order placed in the order book.	order number
Successful	6 - Order partially placed in the order book and partially closed.	order number
Transaction aborted	GEN_CDC_INT_CLOSED - Instrument type is not open for this transaction type.	-
Transaction aborted	MP_MATCH_LOW_VOLUME - Fill or Kill order could not be filled because of low order book size (volume).	-

- The only supported user action on a central inactive order is to delete that inactive order using an MO40

12 MO3 Single Order Update

12.1 Transaction Function

The MO3 transaction is used to amend an existing order entered via MO1 transaction. Only one order can be amended at a time. The order is identified by the order number, series field and the bid/ask flag.

The transaction must always have the series order_number_u, bid_or_ask_c and items_n populated. size_n is not used for this transaction.

Other than the fields to identify the order, the fields in the sub structures should be completed only if they are to be amended. Sub structures only have to be supplied if they contain a field that is to be amended and the sub structures can be defined in any order. Fields that are to be changed should contain the new value required. Fields that are not to be changed are set to be zero.

The following exceptions exist:

- Field minimum_quantity_i in sub structure centre_point_order_t

  If this field is to be amended to zero (i.e. no minimum acceptable quantity for Centre Point Block or Limit Sweep order), it should be set to zero in the MO3 transaction. If the field is not to be amended, it should be set to its current value or the centre_point_order_t sub structure should not be sent at all.

- Field crossing_key_i in sub structure crossing_t

  If this field is to be amended to zero (i.e. no Crossing Key), it should be set to zero in the MO3 transaction. If the field is not to be amended, it should be set to its current value or the crossing_t sub structure should not be sent at all.

- Field exch_order_type_n for Undisclosed orders only

  Undisclosed orders (exch_order_type_n = 32) can be amended to be disclosed (exch_order_type_n = 0). When amending undisclosed orders, exch_order_type_n must be set to 32 unless the order is to be amended to become disclosed.

Note:
This means that the price of an order can never be changed to a market price that is zero. For the same reason, the validity time of an order can never be changed to zero. A zero setting indicates that a field is to be left unchanged in the order book.
For fields that are char arrays, users must complete the field with NULLs to indicate that the field should be ignored.

It is possible to carry out several amendments on the one order at the same time. The following fields may be amended:

- premium_i
- quantity_i
- display_quantity_i
- time_validity_n
- exchange_info_s

- customer_info_s
- give_up_member
- ex_client_s
- open_close_req_c
- minimum_quantity_i
- mid_tick_c
- single_fill_minimum_quantity_c
- exch_order_type_n (Undisclosed orders only)
- crossing_key_i
- regulatory_data_s
- short_sell_quantity_i
- participant_order_attribute_i
- counter_order_attributes_i.

When amending the time validity of an order, ASX Trade will take the new time relative to when the amendment was received. For example, if an order is placed on day one with a time validity of '5:22' (indicating it is valid for 22 days), and then amended on day three to '5:2' (indicating that is valid for only two days), then it will be set to expire at the end of day four (two days after the MO3 transaction).

The order identifier of an amended order does not change, even though the omniapi_tx_ex(…) function returns a different value in its order identifier parameter. This parameter should be ignored on this transaction.

12.1.1 Changes to Price

A change in price will result in the order losing its priority in the market. A change of price can be affected by amending the premium_i field and, for Centre Point, Centre Point Block and Limit Sweep orders, the mid_tick_c field.

12.1.2 Changes to Quantity

An increase in quantity of an order will result in the order losing its priority in the market.

There are two options for amending order quantity; delta and absolute. Delta changes amend the quantity of an order by the given amount, positive to increase the quantity, negative to reduce the quantity. Absolute change means that the quantity should be set to the value in the quantity field.

This is selected by using the field delta_quantity_c. Setting this field to '1' indicates that absolute quantities should be used, setting it to '2' indicates that quantities should be amended by the given delta amount.

The delta_quantity_c setting is applicable for quantity_i, display_quantity_i and short_sell_quantity_i, but does not apply to minimum_quantity_i. The minimum_quantity_i value stated in the centre point order structure is always considered as absolute.Examples:

Original Order	Amendment	Result
quantity_i = 1,000	delta_quantity_c = 1 quantity_i = 600	quantity_i = 600
quantity_i = 1,000	delta_quantity_c = 2 quantity_i = 600	quantity_i = 1,600
quantity_i = 1,000	delta_quantity_c = 2 quantity_i = -600	quantity_i = 400
quantity_i = 10,000	delta_quantity_c = 1	quantity_i = 8,000

Original Order	Amendment	Result
display_quantity_i = 5,000	quantity_i = 8,000 display_quantity_i = 6,000	display_quantity_i = 6,000
quantity_i = 10,000 display_quantity_i = 5,000	delta_quantity_c = 2 quantity_i = -1,000 display_quantity_i = 1,000	quantity_i = 9,000 display_quantity_i = 6,000
quantity_i = 12,000 display_quantity_i = 7,000	delta_quantity_c = 2 quantity_i = -12,000 display_quantity_i = -7,000	Order deleted.
quantity_i = 12,000 display_quantity_i = 7,000	delta_quantity_c = 2 quantity_i = 0 (*no change*) display_quantity_i = -7,000	Order deleted.

12.1.2.1 Balance Quantity

If the field balance_quantity_i is provided, ASX Trade checks this quantity against the existing total quantity of the order prior to applying the amendment. If the two match then the amendment is applied, if not, an error is returned.

12.1.2.2 Minimum Acceptable Quantity

The MAQ for Centre Point Block and Limit Sweep Orders specified in the field minimum_quantity_i can be increased or decreased without the order losing priority in the market.

12.1.2.3 Short Sell Quantity

The partial short quantity of short sell orders specified in the field short_sell_quantity_i can be increased or decreased without the order losing priority in the market.

12.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	The transaction complies with the VIT concept. The topmost struct is single_order_update_t (named structure 34809).
Partitioned	true

This is a variable information transaction. Sub-headers within the message identify what is contained in the message. The overall structure is:

- single_order_update_t (named structure 34809)
- one or more sequences of:
  - sub_item_hdr_t
  - a choice of:
    - basic_order_update_t (named structure 34815)
    - exchange_info_t (named structure 50004)
    - free_text_t (named structure 34801)
    - clearing_info_t (named structure 34802)

- reserve_order_t (named structure 34812)
- centre_point_order_t (named structure 34816)
- enhanced_cp_matching_t (named structure 34831)
- crossing_t (named structure 34820)
- regulatory_t (named structure 34821)
- short_sell_order_t (named structure 34829).

As an example, the sequence of sub-structures for amending a Limit order with a Crossing Key is as follows:

- basic_order_update_t (named structure 34815)
- crossing_t (named structure 34820).

Sub-structures only need to be provided if they contain one or several variables to be amended.

12.3 Message Structures

12.3.1 single_order_update_t (named structure 34809)

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 3}.
series	series_t The series for which the order exists. The series must have a value assigned to every field; it does NOT act as a wildcard whereby some fields can be ignored.
order_number_u	quad_word The identifier of the order that is being amended.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
filler_3_s	char[3] Ignore. Used for byte alignment.
items_n	uint16_t The number of sub-items following this top structure.
size_n	uint16_t Not used for this transaction.

12.3.2 sub_item_hdr_t

Variable	Description
named_struct_n	uint16_t Contains a number, indicating the type of structure that follows.
size_n	uint16_t Not used for this transaction.

12.3.3 basic_order_update_t (named structure 34815)

Variable	Description
premium_i	int32_t The price of the order. A combination of this field and the order_type_c field signify different types of orders. 0 = Market order. Any other value than zero = Limit order (order_type_c = 1 or 65). The price for a TMC order can be positive, zero or negative. When amending an order, setting this field to zero will not make it a Market order. Zero indicates that no changes are to be made to this field.
quantity_i	int64_t Quantity of the order. When amending an order, this field can be set to the new value (delta_quantity_c = 1), or the amount by which it should be changed (delta_quantity_c = 2), or zero to indicate 'no change'. For iceberg orders this the total quantity of the order. The shown quantity portion of the iceberg order is defined in the reserve_order_t sub structure. For iceberg orders, an increase to total quantity is not allowed.
balance_quantity_i	int64_t 0 = No balance check is performed. >0 = Total quantity of the order in the system must match the balance quantity otherwise the transaction will be rejected. <0 = Transaction is rejected, a negative value is not allowed.
time_validity_n	uint16_t This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte). Unit = 0, Count = 0 (i.e. binary = 0): Order is an "Immediate" type order. Fill Or Kill when block_n = 0 or Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not have an expiry date, if this order is for an equity it will be valid for the maximum allowed time for that particular instrument type. Unit = 5, Count = a positive integer (i.e. binary 101 0000 0011, hex 503, dec 1283): Order will be valid for that many calendar days, including today. Expiry will occur at the end of day's trading on the final day. The answer contains the number of days left for the order, decreasing by one every day. It does NOT contain the number of days when the order was originally placed. Unit = 6, Count = 0 (i.e. binary 110 0000 0000, hex 600, dec 1536): Order is "Good Till Cancelled" type. Order will be valid for the maximum allowed time for that particular instrument type. Centre Point and Centre Point Block orders are not carried over to the next day, i.e. they are purged at the end of the day, regardless of the entered time_validity_n. For Centre Point Block orders, validity FoK is only allowed when the MAQ is zero or equal to the order quantity.
exch_order_type_n	uint16_t Exchange specific order types. This field cannot be amended, with one exception; a disclosed order can be amended to

Variable	Description
	become undisclosed and an undisclosed order can be amended to become disclosed. Possible values for an existing disclosed order: 0 = No change 32 = Change to an Undisclosed order. Possible values for an existing Undisclosed order: 0 = Change to a disclosed order 32 = No change. ~~Note: When MO3 is used to amend an OUCH order, it's not allowed to change the order type to 32 (undisclosed order). Attempting to change EXCH_ORDER_TYPE_N to 32 for an OUCH order will result in a reject with code -420339 (ME_MATCH_INS_EXCH_ORDER_TYPE)~~
delta_quantity_c	uint8_t Indicates if the quantity_i reflects absolute quantity or delta quantity. Possible values: 1 = Absolute quantity 2 = Delta quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.

12.3.4 reserve_order_t (named structure 34812)

Variable	Description
display_quantity_i	int64_t The shown quantity of an iceberg order. The reserve_order_t sub structure only needs to be specified when entering iceberg orders. Can also be used for Limit Sweep order types to enter them as an iceberg order.
original_display_quantity_i	int64_t Not used in the MO1 and MO3 transactions. Set to zero.

12.3.5 centre_point_order_t (named structure 34816)

Variable	Description
minimum_quantity_i	int64_t MAQ of Centre Point Block order, Any Price Block order or Limit Sweep order when executing in Centre Point. Specifies the minimum quantity that must be traded in each execution cycle. Amending this field to zero means that the Centre Point Block or Limit Sweep order has no MAQ. If the current MAQ for the order should be retained this field has to be set to its current value. Must be set to zero for Centre Point orders (exch_order_type_n = 64 or 66).
mid_tick_c	uint8_t Specifies whether the limit price of a Centre Point Limit or Centre Point Block Limit order should be a half-tick more aggressive (i.e. improved) and/or allowed for permitted prices other than mid-point ('dark limit' order). Specifies whether a Limit Sweep order is eligible for passive execution in Centre Point at a half-tick above the limit price.

Variable	Description
	1 = mid-tick attribute set on 2 = mid-tick attribute set off 3 = allowed for permitted prices in addition to mid-point (‘dark limit’ order) 4 = allowed for permitted prices in addition to mid-point (‘dark limit’ order), with mid-tick attribute set on. 5 = Any Price Block order (can only be used with Centre Point Block Limit orders). 6 = Any Price Block order with mid-tick attribute set to on (can only be used with centre Point Block Limit orders). Limit Sweep orders fully integrate the liquidity in ASX Centre Point and ASX TradeMatch and will interact with both ‘mid-point only’ and ‘dark limit’ Centre Point and Centre Point Block orders and Any Price Block orders. The only allowed mid_tick_c values for Limit Sweep orders are 1 and 2. Limit Sweep orders cannot be amended to have their mid-tick attribute set to on.
preference_only_c	uint8_t Specifies whether a Centre Point Order or a Centre Point Block order is a Preference and Kill order or not. For Limit Sweep orders, only ‘0’ or ‘2’ are valid values. 0 = not defined or ‘no’, order is not a Preference and Kill order. 1 = ‘yes’, order is a Preference and Kill order. Time validity must be set to Fill and Kill or Fill or Kill for this option. 2 = ‘no’, order is not a Preference and Kill order. This variable cannot be amended.
single_fill_minimum_quantity_c	uint8_t Specifies whether the MAQ (minimum_quantity_i) of Centre Point Block or Limit Sweep orders must be satisfied in a single fill or not. 0 = not defined 1 = minimum acceptable quantity must be satisfied in a single fill 2 = minimum acceptable quantity may be satisfied in multiple fills (aggregated execution). Must be set to 0 or 2 for Centre Point orders (exch_order_type_n = 64 or 66). Can only be set to 1 for Centre Point Block and Limit Sweep orders that have a minimum_quantity_i > 0.
filler_1_s	char[1] Ignore. Used for byte alignment.

12.3.6 exchange_info_t (named structure 50004)

Variable	Description
exchange_info_s	char[32] A free text field used at the participant’s discretion. **Note**: When entering Centre Point and Sweep orders it is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.

12.3.6.1 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.

Variable	Description
boq_list_s	char[6] Ignore. Not used when amending orders with MO3.
initial_trd_report_c	uint8_t Ignore. Not used when amending orders with MO3.
filler_1_s	char[1] Ignore. Used for byte alignment.
extended_price_q	int64_i This field may be set to a trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000) for Centre Point orders trading at an extended price, or the special value indicating that it is to be ignored. If the 63rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

12.3.7 free_text_t (named structure 34801)

Variable	Description
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.
filler_1_s	char[1] Ignore. Used for byte alignment.

12.3.8 clearing_info_t (named structure 34802)

Variable	Description
give_up_member	give_up_member_t The clearing identifier used for the order. See give_up_member_t below.
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
open_close_req_c	uint8_t Set to Zero
filler_1_s	char[1] Ignore. Used for byte alignment.

12.3.8.1 give_up_member_t (named structure 50002)

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to 'AU', indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the Clearing Participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

12.3.9 crossing_t (named structure 34820)

Variable	Description
crossing_key_i	int32_t Crossing key for Unintentional Crossing Prevention. When two orders from the same participant with the same crossing key trade out, the resulting trade is treated like a booked transaction and not published to the market as a trade. Setting this field to zero for an order means “no Unintentional Crossing Prevention” for this order.

12.3.10 regulatory_t (named structure 34821)

Variable	Description
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.

12.3.10.1 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False.
execution_venue_s	char[4]	2 to 5	Execution venue. Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

12.3.11 short_sell_order_t (named structure 34829)

Variable	Description
short_sell_quantity_i	int64_t Partial short quantity of a short sell order. For orders that are not short sell orders (2nd least significant bit not set), this sub-structure should not be included at all. For orders that are short sell orders (2nd least significant bit is set), must be equal to or less than the total order quantity and greater than zero to amend short sell quantity. If set to zero this will be treated as ‘no change’ from the previous short sell quantity.

Variable	Description
	When reducing quantity for a short sell order, if delta_quantity_c =2, then the short sell quantity must be decremented (e.g. -100), to ensure that short sell quantity is equal to or less than remaining quantity.

12.3.12 enhanced_cp_matching_t (named structure 34831)

This optional sub-structure is reserved for future use with Centre Point and Sweep orders. Functionality relating to these fields will be detailed at a later stage. Until then, this sub-structure should not be used.

Variable	Description
participant_order_attribute_i	uint32_t Currently not supported
order_attributes_i	uint32_t Currently not supported

12.3.13 Return Codes

After a successful MO3 transaction, the quantity of the order prior to the change is returned in the transaction parameter of the omniapi_tx_ex(…) function.

Note:
Not changing anything at all, as well as attempting to change fields that cannot be amended may be considered a successful operation from the status returned by the function call. However the order will be unchanged.

An MO3 transaction may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted is returned to the sender.

Cstatus	Txstat	Description
Successful	n	Order quantity before the amendment.
Transaction aborted	GEN_CDC_INT_CLOSED	Instrument type is not allowed for this transaction type.
Transaction aborted	GEN_MATCH_INV_ALTER	Amendment is not allowed with retained priority.
Transaction aborted	GEN_MATCH_ORD_NOT_FOU	The specified order to be amended was not found.

13 MO4 Delete Order

13.1 Transaction Function

This transaction is used to remove one or more orders from the order book. This transaction can affect several orders at once; i.e. users can specify to delete a group of orders.

If one specific order is to be deleted, the following fields must be specified:

- series (must be fully completed)
- order_number_u
- bid_or_ask_c.

When a group of orders is to be deleted, the group is defined by the following fields:

- series

This can be completed either as underlying (Country Number plus Market Code plus Commodity Code) or as instrument class.

- whose

This is used to specify My, Our, My Client's or Our Client's orders and consists of the following:

- Customer (ex_customer_s)
- User (user_id_s)
- Client (ex_client_s).
- bid_or_ask_c

Set to Bid, Ask or both Bid and Ask.

13.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	delete_trans_t
Partitioned	true

13.3 Message Structure

13.3.1 delete_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 4}.
series	series_t The series for which the order exists. When deleting a single order all the fields must be filled in.

Variable	Description
	When deleting a group of orders, user can specify an underlying (accompanied with Country Code and Market Code) or a whole instrument class.
order_number_u	quad_word The order identifier of the order to be deleted, or set to zero when deleting a group of orders.
whose	whose_t Used as a filter for a group of orders to delete. Users should zero-fill the structure if they are only deleting one specific order. See whose_t sub structure below.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Bid (when deleting a specific order or a group of orders) 2 = Ask (when deleting a specific order or a group of orders) 0 = Bid and Ask (when deleting a group of orders).
customer_info_s	char[15] Customer information - a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] A free text field used at the participant's discretion.

13.3.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters for orders to be deleted. It can be configured to specify My, Our, My Client's or Our Client's orders.

Type of Order	Fields to be Completed
My orders	country_id_s, ex_customer_s, user_id_s
Our orders	country_id_s, ex_customer_s
My orders for a specific client	country_id_s, ex_customer_s, user_id_s ex_client_s
Our orders for a specific client	country_id_s, ex_customer_s and ex_client_s

Note:
Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, the client field may contain the wildcard character '*' (substitutes zero or more characters) or '%' (substitutes a single character).

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] Client - free text field typically used to indicate to the participant the ultimate client making the order.

Variable	Description
filler_2_s	char[2] Ignore. Used for byte alignment.

13.3.3 trading_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For Trading participants this is typically a three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of ASX Trade users. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

13.3.4 Examples

Example 1

To delete all “My” orders for European Call Options over XPJ on the bid side:

- series_t is completed with:
  - country_c = 15
  - market_c = 1
  - instrument_group_c = 1 and
  - commodity_n = 35022
- whose_t is completed with
  - ex_customer_s = <<My Participant ID>>
  - user_id_s = <<My User ID>>
  - bid_or_ask_c = 1.

Example 2

To delete all “My Company’s” orders for the client “JoeBloggs” involving BHP in Equity Market Group 1 (A-B):

- series_t is completed with:
  - country_c = 15
  - market = 101
  - commodity_n = 5080
- whose_t is completed with

- ex_customer_s = << My Participant ID>>
- ex_client_s = “JoeBloggs”

- bid_or_ask_c = 0.

13.3.5 Return Codes

Note:
Not finding an order to delete is considered a successful operation.

If deleting a group of orders, then the number of orders deleted will be returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates no orders were deleted.

If deleting a single order then the quantity of that order is returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates the order was not deleted.

A MO4 transaction may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted will be returned to the sender.

Cstatus	Txstat
Successful	When deleting a group of orders, the two least significant bytes in the field specify the number of orders deleted; a zero indicates no orders were deleted. The two most significant bytes in the field specify the number of orders that should have been deleted but still remain in the order book due to market constraints.
Successful	When deleting a single order, this field indicates the order quantity before deletion; a zero indicates the order was not deleted.

14 MO36 Two Sided Price Quotation Block Entry

14.1 Transaction Function

This transaction is used for entering, replacing or deleting up to a configurable maximum number of two-sided quotations in the order book. The maximum number of quotes that can be placed in one transaction is retrieved by using the Maximum Block Order Sizes query (refer to *MQ99* in *ASX Trade Queries*). The transaction is rejected, if the maximum limit is exceeded.

Users should take care when less than 100 items are used in the array. The omniapi_tx_ex(…) function requires the query buffer size to prefix the query buffer. When less than 100 items are used in the array this size is not the result of size (block_price_trans_t). Rather it is the size of the fields in the structure with the addition of the size of only the used items in the array.

The transaction allows the inclusion of an order number and thereby replaces existing quotes with new ones. If the previous quotes do not exist (e.g. they were traded before a subsequent transaction could be placed) the new quotes will still be placed into the order book.

Be aware that:

- It is not possible to have more than one bid quote and one ask quote per series in the transaction. A series can only be entered once in the transaction.
- The bid order to be replaced is specified by order_number_bid_u and series fields. The ask order to be replaced is specified by order_number_ask_u and series fields. To replace the whole two-sided quote, specify the order number for the bid side and the order number for the ask side together with series.
- By setting bid/ask total volume to zero, the order is sent as a normal order without hidden quantity.
- When the bid/ask quantity and the bid/ask total volume are different, the value entered for the bid/ask quantity will be the shown quantity in the order book and the bid/ask total volume will show the total quantity for the order. When the shown quantity has traded out, it is refreshed from the total quantity, as per normal iceberg order behaviour. Some instruments may be configured to not allow iceberg orders, in which case a quote with total volume > shown quantity would be rejected.
- By setting both the bid/ask quantity and bid/ask total volume to zero, the previous quote is deleted and not replaced by a new one.
- When replacing a quote, setting the time validity to 0 is not supported. Setting the time validity to 0 when replacing a quote will result in a reject.

Note:
The set of series used in one transaction cannot go across more than one ME partition. All series must exist in the same ME partition.
The MO36 transaction does not handle combinations.
Market Making applications must utilise the MO36 for entering and amending their quotes.
Market Making applications must not be configured to requote their entire spread in a market when only a single side of their previous quote is affected by a movement in the associated underlying. Market Makers must only requote the affected bid or ask side of their spread.

14.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0

Struct Name	block_price_trans_t
Partitioned	true

14.3 Message Structure

14.3.1 block_price_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 36}.
series	series_t The series must be completed for MO36 transactions. It is mandatory to fill in the series and it has to be set to any of the series contained in the quotation block structure.
give_up_member	give_up_member_t Indicates the clearing identifier for this quote. See give_up_member_t sub structure below.
exchange_info_s	char[32] A free text field used at the participant's discretion.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. The same data will apply to all quotes within this transaction. See ASX specific overlay of regulatory_data_s variable below.
item_c	uint8_t The number of items populated in the array below.
item	block_price_trans_item_t[14] Array of items – maximum 14 items. See block_price_trans_item_t sub structure.

14.3.1.1 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the clearing participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

14.3.1.2 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N = False
execution_venue_s	char[4]	2 to 5	Execution venue. Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

14.3.2 block_price_trans_item_t

Variable	Description
series	series_t The series for which this quote is entered, replaced or deleted.
order_number_bid_u	quad_word The order identifier for this item of the block transaction on the bid side. This field is only used if the user is replacing an existing quote on the bid side.
order_number_ask_u	quad_word The order identifier for this item of the block transaction on the ask side. This field is only used if the user is replacing an existing quote on the ask side.
bid_premium_i	int32_t The price of the quote on the bid side of this item in a block transaction. A combination of this field and the order_type_c field signify different types of orders. 0 = Market order (order_type_c > 1) Any value = Fixed price order (order_type_c = 1).
ask_premium_i	int32_t The price of the quote on the ask side of this item in a block transaction. A combination of this field and the order_type_c field signify different types of orders. 0 = Market order (order_type_c > 1) Any value = Fixed price order (order_type_c = 1).
bid_quantity_i	int64_t Indicates the shown quantity of the quote on the bid side of this item in a block transaction.

Variable	Description
	Note: It is possible for an order amendment to have a negative quantity with the delta_quantity_c field set to two, indicating that this value should be subtracted from the existing order quantity.
ask_quantity_i	int64_t Indicates the shown quantity of the quote on the ask side of this item in a block transaction. **Note**: It is possible for an order amendment to have a negative quantity with the delta_quantity_c field set to two, indicating that this value should be subtracted from the existing order quantity.
bid_total_volume_i	int64_t Indicates the total quantity of the quote on the bid side of this item in a block transaction.
ask_total_volume_i	int64_t Indicates the total quantity of the quote on the ask side of this item in a block transaction.
block_n	uint32_t Block size. Possible values: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	uint16_t The time validity that applies to both quotes in this item - on the bid and the ask side. This field is made up of two eight bit parts - unit (most significant byte) and count (less significant byte). Unit = 0, Count = 0 (i.e. binary = 0): Order is an “Immediate” type order. Fill Or Kill when block_n = 0. Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not expire then if this order relates to an equity it will be valid for the maximum expiry time for an order. Unit = 5, Count = a positive integer (i.e. binary 101 0000 0011, hex 503, dec 1283): Order will be valid for that many calendar days, including today. Expiry will occur at the end of day’s trading on final day. Unit = 6, Count = 0 (i.e. binary 110 0000 0000, hex 600, dec 1536): Order is “Good Till Cancel” type. Order will be valid for the maximum allowed time for that particular instrument type.
order_type_c	uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0, time_validity_n != 0).
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
delta_quantity_c	uint8_t How to treat both the bid and ask quantity fields when an existing quote is being replaced: Possible values include:

Variable	Description
	1 = Absolute quantity 2 = Delta quantity.
filler_2_s	char[2] Ignore. Used for byte alignment.

14.3.3 Return Codes

After a successful MO36 transaction, an order number and the number of two-sided quotations entered successfully are returned to the sender. The same order number is assigned to all successfully placed quotes in the block transaction.

Market Makers (refer to *ASX Trade Open Interface Function Calls* for more information) are the only users allowed to access this transaction. Firm Order Book broadcasts (refer to *BO5 Firm Order Book* in *ASX Trade Broadcasts*) are not disseminated on the placement of two-sided quotes. The BO5 broadcast is issued only when the quotes trade or are deleted by the user.

Therefore Market Makers will be sent the Block Transaction Response broadcast (refer to *BO99 Block Transaction Response* in *ASX Trade Broadcasts)* informing which quotes failed and the corresponding error code.

Note:
If all quotes in the block transaction are rejected then the BO99 is not sent.

Note:
The system deserialises the Item Array based on the number of items specified in the MO36 request. For example, if the item is 1 and there are 2 blocks in the request, only the first Item Array is accepted. Please ensure that the item_c reflects the number of block_price_trans_item_t items.

An MO36 transaction may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted is returned to the sender.

Cstatus	Txstat	ordidt
Successful	Number of two sided quotations successfully entered and/or matched.	order number

15 MO37 Two-Sided Price Quotation

15.1 Transaction Function

The MO37 transaction is used to add, replace or delete a single two-sided quotation in the order book.

15.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	hv_price_2_trans_t
Partitioned	true

15.3 Message Structure

15.3.1 hv_price_2_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 37}.
series	series_t The series for which this quote is entered, replaced or deleted.
give_up_member	give_up_member_t Indicates the clearing identifier for this quote. See give_up_member_t sub structure below.
order_number_bid_u	quad_word The order identifier for the bid side. This field is only used if the user is replacing an existing quote on the bid side.
order_number_ask_u	quad_word The order identifier for the ask side. This field is only used if the user is replacing an existing quote on the ask side.
bid_premium_i	int32_t The price of the quote on the bid side.
ask_premium_i	int32_t The price of the quote on the ask side.
bid_quantity_i	int64_t Indicates the shown quantity of the quote on the bid side.
ask_quantity_i	int64_t Indicates the shown quantity of the quote on the ask side.
bid_total_volume_i	int64_t Indicates the total quantity of the quote on the bid side of this item.
ask_total_volume_i	int64_t Indicates the total quantity of the quote on the ask side of this item.
block_n	uint32_t Block size. Possible values:

Variable	Description
	0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	uint16_t This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte). Unit = 0, Count = 0 (i.e. binary = 0): Order is an “Immediate” type order. Fill Or Kill when block_n = 0. Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256):Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not expire then if this order relates to an equity it will be valid for the maximum expiry time for an order. Unit = 5, Count = a positive integer (i.e. binary 101 0000 0011, hex 503, dec 1283): Order will be valid for that many calendar days, including the day of placement. Expiry will occur at the end of day’s trading on final day. Unit = 6, Count = 0 (i.e. binary 110 0000 0000, hex 600, dec 1536): Order is “Good Till Cancel” type. Order will be valid for the maximum allowed time for that particular instrument type.
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
order_type_c	uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values: 1 = Limit price order (premium_i = an integer) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0, time_validity_n != 0).
customer_info_s	char[15] Customer information - a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] A free text field used at participant’s discretion.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.

15.3.1.1 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the Clearing Participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1]

Variable	Description
	Ignore. Used for byte alignment.

15.3.1.2 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False
execution_venue_s	char[4]	2 to 5	Execution venue Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

15.3.2 Return Codes

Cstatus	Txstat		ordidt
Successful	Bit 0 set	No part of the ask order placed in the order book and no part closed.	order number
Successful	Bit 1 set	The whole ask order closed.	order number
Successful	Bit 0 and Bit 1 set	The ask order partially closed and nothing placed in the order book.	order number
Successful	Bit 1 and 2 set	The ask order partially placed in the order book and partially closed.	order number
Successful	Bit 2 set	The whole ask order placed in the order book.	order number
Successful	Bit 5 set	No part of the bid order placed in the order book and no part closed.	order number
Successful	Bit 5 and 6 set	The bid order partially closed and nothing placed in the Order book.	order number
Successful	Bit 6 set	The whole bid order closed.	order number
Successful	Bit 6 and 7 set	The bid order partially placed in the order book and partially closed.	order number
Successful	Bit 7 set	The whole bid order placed in the order book.	order number
Transaction aborted	GEN_CDC _INT_CLOSED Instrument type not open for this transaction type.		

Cstatus	Txstat	orddt
Transaction aborted	MP_MATCH_LOW_VOLUME Fill or Kill order could not be filled because of low order book volume.	

16 MO40 Delete Inactive Order

16.1 Transaction Function

This transaction is used to delete one or more central inactive orders. This transaction can affect several inactive orders at once, i.e. users can specify to delete a group of inactive orders.

If one specific order is to be deleted the following fields must be specified:

- series (must be fully completed)
- order_number_u
- bid_or_ask_c.

When a group of inactive orders is to be deleted, the group is defined by the following fields:

- series_t

  Can be completed either as underlying (Country Code plus Market Code plus Commodity Code) or as instrument class.

- whose_t

  Is used to specify My, Our, My Client's or Our Client's orders and consists of the following:

  - Customer (ex_customer_s)
  - User (user_id_s)
  - Client (ex_client_s)

- bid_or_ask_c

  - Set to Bid, Ask or both Bid and Ask.

16.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	delete_trans_t
Partitioned	true

16.3 Message Structure

16.3.1 delete_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 40}.
series	series_t

Variable	Description
	The series for which the order exists. When deleting a single order all the fields must be filled in. When deleting a group of orders, the user can specify an underlying (accompanied with Country and Market) or a whole instrument class.
order_number_u	quad_word The identity of the order to be deleted, or set to zero when deleting a group of orders.
whose	whose_t Used as a filter for a group of orders to delete. Users should zero-fill the structure if they are only deleting one specific order. See whose_t sub structure below.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Bid (when deleting a specific order or a group of orders) 2 = Ask (when deleting a specific order or a group of orders) 0 = Bid and Ask (when deleting a group of orders).
customer_info_s	char[15] Customer information - a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] A free text field used at participant's discretion.

16.3.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters for orders to delete. It can be configured to specify My, Our, My Client's or Our Client's inactive orders.

Type of Order	Fields to be Completed
My orders	country_id_s, ex_customer_s, user_id_s
Our orders	country_id_s, ex_customer_s
My orders for a specific client	country_id_s, ex_customer_s, user_id_s ex_client_s
Our orders for a specific client	country_id_s, ex_customer_s and ex_client_s

Note:
Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, the client field may contain the wildcard character '*' (substitutes zero or more characters) or '%' (substitutes a single character).

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] Client - free text field typically used to indicate to the user the ultimate client making the order.

Variable	Description
filler_2_s	char[2] Ignore. Used for byte alignment.

16.3.3 trading_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of an ASX Trade user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

16.3.4 Examples

16.3.4.1 Example 1

To delete all “My” inactive orders for European Call Options over XPJ on the bid side:

- series_t is completed with:
  - country_c = 15
  - market_c = 1
  - instrument_group_c = 1 and
  - commodity_n = 35022
- whose_t is completed with:
  - ex_customer_s = <<My Participant ID>>
  - user_id_s = <<My User ID>>
- bid_or_ask_c = 1.

16.3.4.2 Example 2

To delete all “My Company’s” inactive orders for the client “JoeBloggs” involving BHP in Equity Market Group 1 (A-B):

- series_t is completed with:
  - country_c = 15
  - market = 101
  - commodity_n = 5080
- whose_t is completed with

- ex_customer_s = << My Broker ID>>
- ex_client_s = “JoeBloggs”

- bid_or_ask_c = 0.

16.3.5 Return Codes

Note:
Not finding an inactive order to delete is considered a successful operation.

If deleting a group of inactive orders then the number of orders deleted will be returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates no orders were deleted.

If deleting a single order then the quantity of that order is returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates the order was not deleted.

A MO40 transaction may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted will be returned to the sender.

Cstatus	Txstat
Successful	When deleting a group of orders, the two least significant bytes in the field specify the number of orders deleted; a zero indicates no orders were deleted. The two most significant bytes in the field specify the number of orders that should have been deleted but still remain in the order book due to market constraints.
Successful	When deleting a single order, this field indicates the order quantity before deletion; a zero indicates the order was not deleted.

17 MO74 Delete Unmatched Trade Report

17.1 Transaction Function

This transaction removes single sided trade reports that have not yet matched. It can be used to delete one or more unmatched trade reports.

If one specific unmatched trade report is to be deleted, the following fields must be specified:

- series (must be fully completed)
- order_number_u
- bid_or_ask_c.

When a group of unmatched trade reports is to be deleted, the group is defined by the following fields:

- series_t

  Can be completed either as underlying (Country Code plus Market Code plus Commodity Code) or as instrument class.

- whose_t

  Is used to specify My, Our, My Client's or Our Client's unmatched trade reports and consists of the following:

  - Customer (ex_customer_s)
  - User (user_id_s)
  - Client (ex_client_s)

- bid_or_ask_c

  - Set to Bid, Ask or both Bid and Ask.

17.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	delete_trans_t
Partitioned	false

17.3 Message Structure

17.3.1 delete_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 74}.
series	series_t

Variable	Description
	The series for which the trade report exists. When deleting a single trade report all the fields must be filled in. When deleting a group of trade reports, the user can specify an underlying (accompanied with Country and Market) or a whole instrument class.
order_number_u	quad_word The identifier of the trade report to be deleted, or set to zero when deleting a group of orders.
whose	whose_t Used as a filter for a group of trade reports to delete. Users should zero-fill the structure if they are only deleting one specific trade report. See whose_t sub structure below.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Bid (when deleting a specific trade report or a group of trade reports) 2 = Ask (when deleting a specific trade report or a group of trade reports) 0 = Bid and Ask (when deleting a group of trade reports).
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] A free text field used at the participant's discretion.

17.3.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters for trade reports to delete. It can be configured to specify My, Our, My Client's or Our Client's unmatched trade reports.

Note:
Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, the client field may contain the wild card character '*' (substitutes zero or more characters) or '%' (substitutes a single character).

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] Client - free text field typically used to indicate to participants the ultimate client making the order.
filler_2_s	char[2] Ignore. Used for byte alignment.

17.3.3 trading_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.

Variable	Description
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information vendors have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

17.3.4 Return Codes

Note:
Not finding an unmatched trade report to delete is considered a successful operation.

If deleting a group of trade reports then the number of trade reports deleted will be returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates no trade reports were deleted.

If deleting a single trade report then the quantity of that trade report is returned in the transaction status field of the omniapi_tx_ex function. A zero value indicates the trade report was not deleted.

A MO74 transaction may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted will be returned to the sender.

Cstatus	Txstat
Successful	When deleting a group of trade reports, the two least significant bytes in the field specify the number of orders deleted; a zero indicates no trade reports were deleted. The two most significant bytes in the field specify the number of trade reports that should have been deleted but still remain in the order book due to market constraints.
Successful	When deleting a single order, this field indicates the trade report quantity before deletion; a zero indicates the trade report was not deleted.

18 MO75 Trade Report

18.1 Transaction Function

This transaction is used to enter a single-sided trade report. For more information on trade reporting refer to *Trade Reporting* in *ASX Trade Introduction and Business Information*.

The trade report sent in the transaction will be matched with a specific counter transaction entered by the counterparty to the trade.

The two reported sides of a trade report will match if:

- The transaction type is equal
- The series is equal
- The quantity is equal
- The price (including the extended price) is equal
- The trade report type is equal
- The as of date is equal
- The settlement date is equal
- The basis of quotation is equal
- One side is a buy and the other side is a sell
- Both have the other participant as its counterpart.

Note:
On this transaction the exchange_info_s is not a free text field. It is overlaid with structure asx_exchange_info_t.

18.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	trade_report_1_trans_t
Partitioned	true

18.3 Message Structure

18.3.1 trade_report_1_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 75}.
series	series_t This is the series of the trade report.
order_var	order_var_t See order_var_t sub structure below.
party	party_t This is the declared counterparty for this part of the trade. See party_t sub structure below.

Variable	Description
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.
give_up_member	give_up_member_t Indicates the clearing identifier for this side of the trade report. See give_up_member_t sub structure.
settlement_date_s	char[8] The settlement date for the trade. Format: YYYYMMDD. If set to null or spaces, the system will calculate a settlement date based on the configured standard settlement period, e.g. (T+2). Where a time_of_agreement_date_s in the past is entered which would mean the calculated settlement date falls on T or a past date, the settlement date is set to T+1.
time_of_agreement_date_s	char[8] The as of date of the trade. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] This field to be left blank.
deferred_publication_c	uint8_t Ignore. Not used by ASX Trade.
filler_1_s	char[1] Ignore. Used for byte alignment.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.
short_sell_quantity_i	int64_t For short sell trade reports, quantity (partial or whole) that is short. For trade reports that are not short sells (exch_order_type_n != 2), must be set to zero. For trade reports that include short sell quantity (exch_order_type_n = 2), must be greater than zero.

18.3.2 order_var_t

Variable	Description
mp_quantity_i	int64_t The quantity of the trade report.
premium_i	int32_t The price of the reported trade, given to the correct number of decimal points as per the configuration of the instrument class. Users can provide up to four decimal points on the price by restating the whole price in the extended_price_q field in the asx_exchange_info_t struct.
block_n	uint32_t Block size. Set to 1.
time_validity_n	uint16_t This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte).

Variable	Description
	Unit = 0, Count = 0 (i.e. binary = 0): Order is an “Immediate” type order. Fill Or Kill when block_n = 0. Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not expire then if this order relates to an equity it will be valid for the maximum expiry time for an order. Unit = 5, Count = a positive integer (i.e. binary 101 0000 0011, hex 503, dec 1283): Order will be valid for that many calendar days, including today. Expiry will occur at the end of day’s trading on final day. Unit = 6, Count = 0 (i.e. binary 110 0000 0000, hex 600, dec 1536): Order is “Good Till Cancel” type. Order will be valid for the maximum allowed time for that particular instrument type.
exch_order_type_n	uint16_t Exchange specific order types. Possible values: 0 = Ignore 2 = Short Sell order.
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information - a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Set to Zero
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible values are listed in *Trade Report Types* in *ASX Trade Introduction and Business Information*.
order_type_c	uint8_t Order type. 1 = Limit price order (premium_i = an integer).
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

18.3.3 party_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5]

Variable	Description
	This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information vendors have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
filler_1_s	char[1] Ignore. Used for byte alignment.

18.3.4 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the clearing participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

18.3.5 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.
boq_list_s	char[6] List of up to three basis of quotations. A basis of quotation is a two character corporate action code.
initial_trd_report_c	uint8_t Indicates if the trade report is the initial part of an Initial or Delayed Trade Report. Possible values include: 0 = No value 1 = Initial trade report 2 = No initial trade report.
filler_1_s	char[1] Ignore. Used for byte alignment.
extended_price_q	int64_i This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the special value indicating that it is to be ignored. If the 63rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

18.3.6 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N = False
execution_venue_s	char[4]	2 to 5	Execution venue Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

19 MO76 Two Sided Trade Report

19.1 Transaction Function

This transaction is used to send a two sided trade report for an off market trade. For more information on trade reporting refer to *Trade Reporting* in *ASX Trade Introduction and Business Information*.

Note:
On this transaction the exchange_info_s is not a free text field. It is overlaid with a structure specific to ASX.

19.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	trade_report_2_trans_t
Partitioned	true

19.3 Message Structure

19.3.1 trade_report_2_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 76}.
series	series_t The series for which the trade is reported.
mp_quantity_i	int64_t Indicates the quantity of the trade report.
premium_i	int32_t Indicates the price of the trade report.
block_n	uint32_t Ignore. Currently not used.
settlement_date_s	char[8] The settlement date for the trade. Format: YYYYMMDD. If set to null or spaces, the system will calculate a settlement date based on the configured standard settlement period, e.g. (T+2). Where a time_of_agreement_date_s in the past is entered which would mean the calculated settlement date falls on T or a past date, the settlement date is set to T+1.
time_of_agreement_date_s	char[8] The as of date. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] This field to be left blank.
ext_t_state_c	uint8_t Trade report code. Possible values are listed under *Trade Report Types* in *ASX Trade Introduction and Business Information*.

Variable	Description
deferred_publication_c	uint8_t Ignore. Not used by ASX Trade.
bid_side	trd_rpt_cust_t Details of the bid side of the trade report. See trd_rpt_cust_t sub structure.
ask_side	trd_rpt_cust_t Details of the ask side of the trade report. See trd_rpt_cust_t sub structure.

19.3.2 bid_side (trd_rpt_cust_t)

Variable	Description
party	party_t This is the declared counterparty for the bid side of the trade. See party_t sub structure below.
ex_client_s	char[10] Client - Free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure below.
open_close_req_c	uint8_t Set to Zero
exch_order_type_n	uint16_t Exchange specific order types. Possible values: 0 = Ignore.
give_up_member	give_up_member_t Indicates the clearing identifier for this side of the trade report. See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.
short_sell_quantity_i	int64_t Short Sell Quantity. Must be set to zero for the bid side of trade reports.

19.3.3 ask_side_t (trd_rpt_cust_t)

Variable	Description
party	party_t This is the declared counterparty for the ask side of the trade. See party_t sub structure below.

Variable	Description
ex_client_s	char[10] Client - free text field typically used to indicate to the user the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure below.
open_close_req_c	uint8_t Set to Zero
exch_order_type_n	uint16_t Exchange specific order types. Possible values: 0 = Ignore 2 = Short Sell order.
give_up_member	give_up_member_t Indicates the clearing identifier for this side of the trade report. See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.
short_sell_quantity_i	int64_t For short sell trade reports, quantity (partial or whole) that is short. For trade reports that are not short sells (exch_order_type_n != 2), must be set to 0. For trade reports that include short sell quantity (exch_order_type_n = 2), must be greater than 0.

19.3.4 party_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
filler_1_s	char[1] Ignore. Used for byte alignment.

19.3.5 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.

Variable	Description
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the clearing participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

19.3.6 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.
boq_list_s	char[6] List of up to three basis of quotations. A basis of quotation is a two character corporate action codes.
initial_trd_report_c	uint8_t Indicates if the trade report is the initial part of an Initial or Delayed Trade Report. Possible values include: 0 = No value 1 = Initial trade report 2 = No initial trade report.
filler_1_s	char[1] Ignore. Used for byte alignment.
extended_price_q	int64_i This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the undefined value indicating that it is to be ignored (0x8000:0000: 0000:0000).

19.3.7 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False
execution_venue_s	char[4]	2 to 5	Execution venue Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.

Variable	Description	Character Position	ASIC defined content
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

19.3.8 Return Codes

Cstatus	Txstat	Description
Successful	2	Transaction Successful

20 MO77 Combination Trade Report

20.1 Transaction Function

This transaction is used to report a two sided combination off-market trade. For more information on trade reporting refer to *Trade Reporting* in *ASX Trade Introduction and Business Information.*

Note:
On this transaction the exchange_info_s is not a free text field. It is overlaid with a structure specific to ASX.

20.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	combo_trade_report_trans_t
Partitioned	true

20.3 Message Structure

20.3.1 combo_trade_report_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'O', 77}.
series	series_t This must be set to the instrument series of the first leg of the combination trade report.
ext_t_state_c	uint8_t Trade report code. Possible values are listed under *Trade Report Types* in *ASX Trade Introduction and Business Information*.
filler_1_s	char[1] Ignore. Used for byte alignment.
items_n	uint16_t Indicates the number of items in the array.
combo_trade_report_trans_item	item[4] See combo_trade_report_trans_item_t sub structure below.

20.3.2 combo_trade_report_trans_item_t

Variable	Description
series	series_t The series of this leg of the combination trade report.
mp_quantity_i	int64_t The quantity of this leg of the combination trade report.
premium_i	int32_t The price of this leg of the combination trade report.

Variable	Description
block_n	uint32_t Ignore. Not currently used.
settlement_date_s	char[8] The settlement date of this leg of the combination trade report. Format: YYYYMMDD. If set to null or spaces, the system will calculate a settlement date based on the configured standard settlement period, e.g. (T+2). Where a time_of_agreement_date_s in the past is entered which would mean the calculated settlement date falls on T or a past date, the settlement date is set to T+1.
time_of_agreement_date_s	char[8] The as of date of this leg of the combination trade report. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] This field to be left blank.
deferred_publication_c	uint8_t Ignore. Not used by ASX Trade.
filler_1_s	char[1] Ignore. Used for byte alignment.
bid_side	trd_rpt_cust_t Details of the bid side of this leg of the combination trade report. See trd_rpt_cust_t sub structure.
ask_side	trd_rpt_cust_t Details of the ask side of this leg of the combination trade report. See trd_rpt_cust_t sub structure.

20.3.3 bid_side (trd_rpt_cust_t)

Variable	Description
party	party_t This is the declared counter party for this side of the trade. See party_t sub structure below.
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure below.
open_close_req_c	uint8_t Set to Zero
exch_order_type_n	uint16_t Exchange specific order type. Possible value:

Variable	Description
	0 = Ignore.
give_up_member	give_up_member_t Indicates the clearing identifier for this side of the trade report. See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.
short_sell_quantity_i	int64_t Short Sell Quantity. Must be set to zero for the bid side of trade reports.

20.3.4 ask_side (trd_rpt_cust_t)

Variable	Description
party	party_t This is the declared counterparty for the ask side of the trade. See party_t sub structure below.
ex_client_s	char[10] Client - free text field typically used to indicate to the user the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure below.
open_close_req_c	uint8_t Set to Zero
exch_order_type_n	uint16_t Exchange specific order types. Possible values: 0 = Ignore 2 = Short Sell order.
give_up_member	give_up_member_t Indicates the clearing identifier for this side of the trade report. See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable below.
short_sell_quantity_i	int64_t For short sell trade reports, quantity (partial or whole) that is short. For trade reports that are not short sells (exch_order_type_n != 2), must be set to zero. For trade reports that include short sell quantity (exch_order_type_n = 2), must be greater than zero.

20.3.5 party_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information vendors have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
filler_1_s	char[1] Ignore. Used for byte alignment.

20.3.6 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a user can be retrieved from the clearing_customer_s field in the clearing participant query (DQ55). Single digits are typically used as identifiers and the rest of the field should be space padded.
filler_1_s	char[1] Ignore. Used for byte alignment.

20.3.7 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.
boq_list_s	char[6] List of up to three basis of quotations. A basis of quotation is a two character corporate action code.
initial_trd_report_c	uint8_t Indicates if the trade report is the initial part of an Initial or Delayed Trade Report. Possible values include: 0 = No value 1 = Initial trade report 2 = No initial trade report.
filler_1_s	char[1] Ignore. Used for byte alignment.
extended_price_q	int64_i This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the special value indicating that it is to be ignored. If the 63rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

20.3.8 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_data_s character positions are to be padded by spaces (ASCII 0x20).

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s	char[1]	0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False
execution_venue_s	char[4]	2 to 5	Execution venue Not required on order messages.
intermediary_id_s	char[10]	6 to 15	Intermediary identifier for agency orders and transactions.
order_origin_s	char[20]	16 to 35	Origin or order information for agency orders and transactions.
filler_s	char[8]	36 to 43	Ignore. Used for byte alignment.

20.3.9 Return Codes

Cstatus	Txstat	Description
Successful	2	Transaction Successful

21 UI1 Ready to Trade

21.1 Transaction Function

When a user has logged on and completed all their necessary initialisations, a Ready Status transaction must be sent to ASX Trade. For a list of steps a typical user would do after logging in, refer to *Conducting a Standard Session* in *ASX Trade Introduction and Business Information*.

21.2 Transaction Properties

Function Call	omniapi_tx_ex
Facility	EP0
Struct Name	application_status_t
Partitioned	false

21.3 Message Structure

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'U', 'I', 1}.
series	series_t Ignore. Currently not used. Zero fill the fields.
application_status_i	uint32_t Set to 1.

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®