



ASX Trade Refresh

ASX Trade Broadcasts

June 2025

Table of Contents

1	Introduction.....	9
1.1	Software Distribution Restrictions	9
1.2	Supported Platforms.....	9
1.3	ASX Trade Support	9
1.4	ASX Trade OI Documentation Suite.....	9
1.5	Restrictions	10
1.5.1	Trading Participant Specific Information	10
1.5.2	Broker Service Providers	10
1.6	Version History.....	11
2	Common Structures.....	13
2.1	Unique Identifiers - transaction_type_t and broadcast_type_t.....	13
2.2	Series Structure – series_t	13
3	Administration Messages	15
3.1	BI9 Price Information Heartbeat	15
3.1.1	Broadcast Function	15
3.1.2	Broadcast Properties.....	15
3.1.3	Message Structure	15
3.2	BN1 Network Status.....	15
3.2.1	Broadcast Function	15
3.2.2	Broadcast Properties.....	16
3.2.3	Message Structure	16
3.3	BN4 Network Status.....	17
3.3.1	Broadcast Function	17
3.3.2	Broadcast Properties.....	17
3.3.3	Message Structure	17
4	BD2 Edited Price Information	19
4.1	Broadcast Function	19
4.2	Broadcast Properties.....	19
4.3	Message Structure	19
4.3.1	broadcast_hdr_t.....	19
4.3.2	item_hdr_t.....	20
4.3.3	sub_item_hdr_t	20
4.3.4	market_info_series_t (named structure 33038)	20
4.3.5	market_info_reason_t (named structure 33043).....	20
4.3.6	market_info_base_t (named structure 33034)	20
4.3.7	market_info_asx_t (named structure 33039)	22
4.3.8	market_info_index_t (named structure 33040).....	22
4.3.9	ob_levels_closing_t (named structure 33031)	23
4.3.10	market_info_asx_extended (named structure 33139).....	23
5	BI7 Information Ready.....	24
5.1	Broadcast Function	24

5.2	Broadcast Properties.....	24
5.3	Message Structure	24
5.3.1	info_ready_t.....	24
6	BI41 Instrument Status Info.....	25
6.1	Broadcast Function	25
6.2	Broadcast Properties.....	25
6.3	Message Structure	25
6.3.1	instrument_status_info_t	25
6.3.2	instrument_status_info_item_t	26
7	BI63 Settlement Price Update	28
7.1	Broadcast Function	28
7.2	Broadcast Properties.....	28
7.3	Message Structure	28
7.3.1	settle_price_update_t.....	28
7.3.2	settle_price_update_item_t	28
8	BI73 Undo Information Ready	30
8.1	Broadcast Function	30
8.2	Broadcast Properties.....	30
8.3	Message Structure	30
8.3.1	undo_info_ready_t	30
9	BI81 Market Announcement	31
9.1	Broadcast Function	31
9.1.1	Market Control Message.....	31
9.1.2	Company Announcement	31
9.1.3	Static Line Message.....	31
9.1.4	Notice Received Message	31
9.2	Broadcast Properties.....	31
9.3	Message Structure	32
9.3.1	broadcast_hdr_t.....	32
9.3.2	sub_item_hdr_t	32
9.3.3	message_core_info_t (named structure 35001)	32
9.3.4	message_information_t (named structure 35002)	33
9.3.5	message_information_item_t.....	33
9.3.6	destination_item_t (named structure 35003).....	33
9.3.7	document_url_t (named structure 35004)	34
10	BI741 Instrument Status Broadcast	35
10.1	Broadcast Function	35
10.2	Broadcast Properties.....	35
10.3	Message Structure	35
10.3.1	me_instrument_status_info_t	35
10.3.2	time_spec_t	36
10.3.3	me_instrument_status_info_item_t	36

11	BL51 Price Limit Update.....	37
11.1	Broadcast Function	37
11.2	Broadcast Properties.....	37
11.3	Message Structure	37
11.3.1	limit_change_t	37
11.3.2	limit_change_item_t.....	37
12	BO2 Order Book Changes	38
12.1	Broadcast Function	38
12.1.1	Add.....	38
12.1.2	Delete.....	38
12.1.3	Amend.....	38
12.2	Broadcast Properties.....	39
12.3	Message Structure	39
12.3.1	ob_changes_no_id_t.....	39
12.3.2	changes_t.....	39
12.3.3	order_no_id_t.....	40
13	BO5 Firm Order Book.....	42
13.1	Broadcast Function	42
13.2	Broadcast Properties.....	43
13.3	Message Structure	43
13.3.1	broadcast_hdr_t.....	46
13.3.2	sub_item_hdr_t	46
13.3.3	block_price_trans_t (named structure 34007).....	46
13.3.4	hv_alter_trans_t (named structure 34010).....	49
13.3.5	hv_alter_trans_p_t (named structure 34110).....	50
13.3.6	hv_order_trans_t (named structure 34005)	50
13.3.7	hv_order_trans_p_t (named structure 34105)	51
13.3.8	hv_price_2_trans_t (named structure 34001)	51
13.3.9	multi_order_response_t (named structure 34906).....	53
13.3.10	order_change_combined_t (named structure 34902).....	54
13.3.11	order_change_separate_t (named structure 34903).....	55
13.3.12	order_chg_sep_trans_ack_t (named structure 34919).....	57
13.3.13	order_info_t (named structure 34917)	57
13.3.14	order_t.....	58
13.3.15	order_price_change_t (named structure 34905)	59
13.3.16	order_return_info_t (named structure 34904)	60
13.3.17	order_trade_info_t (named structure 34920).....	60
13.3.18	order_leg_trade_info_t (named structure 34921)	61
13.3.19	order_trade_info_asx_t (named structure 34922).....	62
13.3.20	segment_instance_number_t (named structure 34901)	63
13.3.21	trade_report_1_trans_t (named structure 34021)	63
13.3.22	trade_report_1_trans_p_t (named structure 34119)	64
13.3.23	trade_report_2_trans_t (named structure 34022)	65
13.3.24	exchange_info_t (named structure 50004).....	66
13.3.25	free_text_t (named structure 34801)	66
13.3.26	clearing_info_t (named structure 34802)	67
13.3.27	single_order_insert_t (named structure 34808)	67

13.3.28	single_order_update_t (named structure 34809)	67
13.3.29	basic_order_t (named structure 34810)	68
13.3.30	reserve_order_t (named structure 34812)	69
13.3.31	basic_order_update_t (named structure 34815)	69
13.3.32	centre_point_order_t (named structure 34816)	71
13.3.33	inactive_order_t (named structure 34818)	72
13.3.34	order_submitter_t (named structure 34819)	72
13.3.35	order_owner_t (named structure 34804)	72
13.3.36	ranking_time_t (named structure 34949)	72
13.3.37	crossing_t (named structure 34820)	72
13.3.38	regulatory_t (named structure 34821)	73
13.3.39	short_sell_order_t (named structure 34829)	73
13.3.40	short_sell_order_change_t (named structure 34830)	73
13.3.41	enhanced_cp_matching_t (named structure 34831)	73
13.3.42	ex_user_code_t	73
13.3.43	give_up_member_t	74
13.3.44	order_var_t	74
13.3.45	party_t	76
13.3.46	trading_code_t	76
13.3.47	originator_trading_code_t	76
13.3.48	time_spec_t	77
13.3.49	asx_exchange_info_t	77

14 BO10 Price Information79

14.1	Broadcast Function	79
14.2	Broadcast Properties	79
14.3	Message Structure	79
14.3.1	equil_price_update_t	79
14.3.2	equil_price_update_item_t	79

15 BO14/BO15 Order Book with Levels81

15.1	Broadcast Function	81
15.2	Broadcast Properties	81
15.3	Message Structure	81
15.3.1	broadcast_hdr_t	81
15.3.2	sub_item_hdr_t	82
15.3.3	ob_levels_sequence_number_t (named structure 33001)	82
15.3.4	ob_levels_order_number_t (named structure 33004)	82
15.3.5	ob_levels_total_quantity_t (named structure 33005)	82
15.3.6	ob_levels_price_volumes_t (named structure 33003)	82
15.3.7	ob_levels_price_volumes_item_t	84
15.3.8	ob_levels_no_of_orders_t (named structure 33033)	84
15.3.9	ob_levels_no_of_orders_item_t	85
15.3.10	ob_levels_id_t (named structure 33002)	85
15.3.11	ob_levels_undisclosed_quantity_t (named structure 33041)	86

16 BO38 Market Maker Protection Settings Information87

16.1	Broadcast Function	87
16.2	Broadcast Properties	87

16.3	Message Structure	87
16.3.1	market_maker_protection_info_t	87
16.3.2	trading_code_t	87
17	BO99 Block Transaction Response	88
17.1	Broadcast Function	88
17.2	Broadcast Properties	88
17.3	Message Structure	88
17.3.1	block_order_response_t	88
17.3.2	block_order_response_item_t	88
18	BU28 Central Group Update	90
18.1	Broadcast Function	90
18.2	Broadcast Properties	90
18.3	Message Structure	90
18.3.1	central_group_update_t	90
18.3.2	da28_t	90
19	BU53 Corporate Action Update	92
19.1	Broadcast Function	92
19.2	Broadcast Properties	92
19.3	Message Structure	92
19.3.1	corp_action_update_bu53_t	92
19.3.2	corp_action_update_bu53_item_t	92
19.3.3	da53_t	92
20	BU56 Special Liability Update	94
20.1	Broadcast Function	94
20.2	Broadcast Properties	94
20.3	Message Structure	94
20.3.1	special_liability_update_bu56_t	94
20.3.2	trading_code_t	94
20.3.3	da56_t	95
20.3.4	item struct for da56_t	95
21	BU87 Market Maker Protection Update	96
21.1	Broadcast Function	96
21.2	Broadcast Properties	96
21.3	Message Structure	96
21.3.1	mm_protection_update_t	96
21.3.2	da87_t	96
22	BU120 Delta Underlying Update	98
22.1	Broadcast Function	98
22.2	Broadcast Properties	98
22.3	Message Structure	98
22.3.1	broadcast_segment_hdr_t	98
22.3.2	item_hdr_t	99

22.3.3	sub_item_hdr_t	99
22.3.4	ns_delta_header_t – named struct number 37001.....	99
22.3.5	time_spec_t	100
22.3.6	ns_remove_t – named struct number 37002.....	100
22.3.7	ns_underlying_basic_t (named struct number 37201)	100
22.3.8	ns_fixed_income_t (named struct number 37202)	102
22.3.9	ns_coupon_dates_t (named struct number 37203).....	103
23	BU122 Delta Instrument Class Update	104
23.1	Broadcast Function	104
23.2	Broadcast Properties.....	104
23.3	Message Structure	104
23.3.1	broadcast_segment_hdr_t.....	105
23.3.2	item_hdr_t.....	105
23.3.3	sub_item_hdr_t	105
23.3.4	ns_delta_hdr_t (named structure 37001)	105
23.3.5	time_spec_t	106
23.3.6	ns_remove_t (named structure 37002)	106
23.3.7	ns_inst_class_basic_t (named structure 37101)	106
23.3.8	ns_price_tick_t (named structure 37102)	109
23.3.9	tick_size_t	109
23.3.10	ns_block_size_t (named structure 37103)	110
23.3.11	ns_calc_rule_t (named structure 37104)	110
23.3.12	ns_crossing_rule_t (named structure 37106)	111
23.3.13	ns_inst_class_ext1_t (named structure 37107)	112
24	BU124 Delta Instrument Series Update.....	114
24.1	Broadcast Function	114
24.2	Broadcast Properties.....	114
24.3	Message Structure	114
24.3.1	broadcast_segment_hdr_t.....	115
24.3.2	item_hdr_t.....	115
24.3.3	sub_item_hdr_t	115
24.3.4	ns_delta_hdr_t (named structure 37001)	115
24.3.5	time_spec_t	116
24.3.6	ns_remove_t (named structure 37002)	116
24.3.7	ns_inst_series_basic_t (named structure 37301).....	116
24.3.8	ns_inst_series_basic_single_t (named structure 37302)	118
25	BU126 Combination Series Update	120
25.1	Broadcast Function	120
25.2	Broadcast Properties.....	120
25.3	Message Structure	120
25.3.1	broadcast_segment_hdr_t.....	120
25.3.2	item_hdr_t.....	121
25.3.3	sub_item_hdr_t	121
25.3.4	ns_inst_series_basic_t (named structure 37301).....	121
25.3.5	ns_combo_series_leg_t (named structure 37308).....	122

26	CB15 Directed Trade Broadcast.....	124
26.1	Broadcast Function	124
26.2	Broadcast Properties.....	124
26.3	Message Structure	124
26.3.1	broadcast_hdr_t.....	124
26.3.2	sub_item_hdr_t	125
26.3.3	cl_trade_base_api_t (named structure 3)	125
26.3.4	give_up_member_t.....	128
26.3.5	account_t.....	128
26.3.6	user_code_t	128
26.3.7	asx_exchange_info_t	129
26.3.8	time_spec_t	129
26.3.9	cl_trade_asx_api_t (named structure 65)	129
26.3.10	party_t	130
26.3.11	cl_deal_extended_price_asx_t (named structure 34)	130
26.3.12	cl_deal_extended_price_asx_item_t	131
26.3.13	cl_deal_trade_report_asx_t (named structure 26)	131
26.3.14	cl_deal_trade_report_asx_item_t.....	131
26.3.15	cl_combo_series_asx_t (named structure 42)	131
26.3.16	cl_trade_regulatory_asx_t (named structure 35)	132
26.3.17	cl_trade_short_asx_t (named structure 36).....	132
27	CB16 Trade	133
27.1	Broadcast Function	133
27.2	Broadcast Properties.....	133
27.3	Message Structure	133
27.3.1	broadcast_hdr_t.....	135
27.3.2	sub_item_hdr_t	135
27.3.3	cl_deal_base_t (named structure 24)	135
27.3.4	time_spec_t	135
27.3.5	cl_deal_asx_t (named structure 25).....	136
27.3.6	cl_trade_public_asx_t (named structure 27)	136
27.3.7	participant_t	137
27.3.8	cl_trade_anonymous_asx_t (named structure 28)	137
27.3.9	cl_deal_trade_report_asx_t (named structure 26)	138
27.3.10	cl_deal_trade_report_asx_item_t.....	138
27.3.11	cl_trade_cancel_asx_t (named structure 30).....	139
27.3.12	cl_combo_series_asx_t.....	139
27.3.13	cl_deal_extended_price_asx_t (named structure 34)	139
27.3.14	cl_deal_extended_price_asx_item_t	140
28	MI4 Quote Request with Volume Information	141
28.1	Broadcast Function	141
28.2	Broadcast Properties.....	141
28.3	Message Structure	141
28.3.1	quote_request_vol_info_t	141
28.3.2	user_code_t	142

1 Introduction

ASX Trade broadcasts are used to notify a specific user, all users belonging to a participant or all logged in users of an event or change occurring in ASX Trade. Broadcasts will notify users of, for example:

- Administration events such as heartbeats and network status price and trade statistics updates
- Order book changes
- Trades
- Updates to the own participant's orders
- Changes in trading and instrument session states
- Incoming market announcements
- Updates to instrument reference data and new instruments introduced during the day
- Quote requests.
- For each broadcast, the following information is provided in this document:
 - Broadcast Function - information about the broadcast
 - Broadcast Properties - the facility required and the function call
 - Message Structure - the structure of the message including a list of variables, their type and description.
- Some broadcasts are dependent on the type of instrument or the type of user they apply to. For example, derivatives instruments, such as options and futures will not generate order book update broadcasts, and Market Makers will not receive Firm Order Book broadcasts (BO5) for quotes, unless the quote trades out or is deleted. Further details are provided in the description of each broadcast function.

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 the 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 (DTR), 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. Trading 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.

Trading 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 resulting from Unintentional Crossing Prevention.

Trading Participant Specific Information is not included in messages 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 a trading participant requests, ASX can provide the vendor with:

- All of the trading participant’s specific information as detailed in *Trading Participant Specific Information* above.
- The vendor participant’s service provider can then integrate this information into their dealing/information systems for the trading participant.
- A service provider 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	June 2014	<p>New Values added for omni_netwrksts for the variable status_u, for SR6. Possible values changed to:</p> <ul style="list-style-type: none"> -208002 = OMNI_DU_DATALOST -208004 = OMNI_DU_LINKLOST -208006 = OMNI_DU_SENDERLOST -208008 = OMNI_DU_SENDERRESTARTED.
v1.2	August 2014	<ul style="list-style-type: none"> XT added in the condition code mapping table for item 79 in ASX Signals/Reference Point column.
v2.0	March 2015	<p>The following has been added:</p> <ul style="list-style-type: none"> New variables added to order_trade_info_asx_t (named structure 34922). New sub-structure added - enhanced_cp_matching_t (named structure 34831). centre_point_order_t (named structure 34816) – changes made. New variables added to cl_trade_asx_api_t (named structure 65). New deal source values added to the trade condition code mapping table. New trade types added in the Mapping table.
v2.1	October 2018	<ul style="list-style-type: none"> Updated to new ASX branding Removal of market, instrument group and trade condition code appendices, which are now covered in ASX Trade Markets, Instrument Groups and Trade Condition Codes.
v2.2	November 2018	<ul style="list-style-type: none"> Addition of values for big_attention_u Removal of references to ASX BookBuild
v3.0	September 2019	<ul style="list-style-type: none"> Updated for ASX Trade Refresh <ul style="list-style-type: none"> Removal of references to Agricultural Derivatives market Operating System Support Removal for Sun Solaris Removal of central inactive value from variable inactive_c Updated variable instance_c description Updated variable combo_mark_c description Updated variable asof_time_s description Updated variable created_time_s description
v3.1	October 2019	<ul style="list-style-type: none"> Updated variable block_n description under ob_levels_id_t structure Updated variable ex_customer_s under give_up_member_t Removal of change_reason_c value as 29 = Entering a central inactive order New structure market_info_asx_extended (named structure 33139) added under BD2 Updated new value on open_close_req_c under cl_trade_base_api_t variable.
v3.2	November 2019	<ul style="list-style-type: none"> Updated for ASX Trade Refresh <ul style="list-style-type: none"> Operating System Support for Rhel6.10 x86 (32 and 64 bit) Updated variable orig_series description under cl_trade_base_api_t structure Updated variable match_id_t description under order_trade_info_t, order_leg_trade_info_t and cl_trade_base_api_t structure

Version	Date	Comment
v3.3	January 2020	<ul style="list-style-type: none"> Updated BI9 Price Information Heartbeat description in section 3.1.1 and 3.1.3.1
v3.4	April 2020	<ul style="list-style-type: none"> Updated trade_venue_c variable description in section 26.3.3 under cl_trade_base_api_t (named structure 3) structure Updated BI9 Price Information Heartbeat description in section 3.1.1 Updated block_n variable description in section 27.3.8 under cl_trade_anonymous_asx_t (named structure 28) structure Notation added in section 27.3.9 cl_deal_trade_report_asx_t (named structure 26)
v3.5	October 2020	<ul style="list-style-type: none"> Sequence number description updated in section 26.3.3 and 27.3.3
v3.6	December 2020	<ul style="list-style-type: none"> Updated exchange_order_type_n variable description in section 13.3.19 order_trade_info_asx_t (named structure 34922) Updated created_date_s variable description in section 26.3.3 cl_trade_base_api_t (named structure 3) Updated as_of_date_s variable description in section 26.3.11 cl_deal_extended_price_asx_t (name structure 34) Updated OMNI_DU_DATA_LOST variable description in section 3.2.3.3 Message Text
v3.7	August 2021	Updated inc_id_s char in section 23.3.7
v3.8	June 2022	Updated the description of extended price fields in section 4.3.10
v3.9	July 2022	<ul style="list-style-type: none"> Updated the description of BN1 Broadcast function in section 3.2.1 Updated the description of cl_combo_series_asx_t in section 26.3.15 Updated the description of variable <i>corp_action_code_s</i> in the sections 26.3.12, 26.3.14, 27.3.10 and 27.3.14.
V4.0	June 2025	<ul style="list-style-type: none"> New change reason code “100 = Cancelled due to Connection loss” added to variable change_reason_c in section 12.3.2

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: <ul style="list-style-type: none"> • M = Matching Engine (ME) • C = Clearing (CL) • I = Information (IN) • D = Common Database (C DB) • O = Operation (OP) • L = List Module (LM) • U = Supervision (SU).
server_type_c	char[1] The Server Types describes the type of the operation that the message will generate. Some samples of the letters indicating the server types are: <ul style="list-style-type: none"> • 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.
market_c	uint8_t

Variable	Description
	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 ASX Trade Markets, Instrument Groups and Trade Condition Codes .
instrument_group_c	uint8_t A numerical value indicating the instrument group. Zero can be used to act as a filter or wild card. For an entire list of possible values see ASX Trade Markets, Instrument Groups and Trade Condition Codes .
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 <i>ASX Trade Introduction and Business Information</i> for more information.
commodity_n	uint16_t A numerical value indicating the commodity (underlying). Example values are: <ul style="list-style-type: none"> • 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 <i>DQ122 Query Delta Instrument Class</i> in <i>ASX Trade Queries</i> for more information.

3 Administration Messages

3.1 BI9 Price Information Heartbeat

3.1.1 Broadcast Function

This broadcast allows the participant to detect if ASX Trade (and in effect the gateway and the connecting network) is still operational.

Currently, ASX Trade has been configured to send out this broadcast every eight seconds. This interval reflects a foundational configuration parameter at the central system and can therefore be considered as constant.

By monitoring this broadcast participants can inform their own clients of the availability of the market and their connection to it.

3.1.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	info_heartbeat_t
Information Type	General

3.1.3 Message Structure

3.1.3.1 info_heartbeat_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 9}.
heartbeat_interval_c	uint8_t The interval in seconds between heartbeats sent out.
instance_c	uint8_t Set to one.
tot_instances_c	uint8_t Set to one.
description_s	char[40] A description on the heartbeat.
filler_1_s	char[1] Ignore. Used for byte alignment.

3.2 BN1 Network Status

3.2.1 Broadcast Function

Network broadcasts relate information on the status of the network, particularly the link between the gateway and ASX Trade. They differ from central server broadcasts as they are disseminated from the gateway, not the central server. It is not necessary to subscribe in order to receive BN1 broadcasts.

3.2.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	omni_broadcast
Information Type	Not Applicable

3.2.3 Message Structure

3.2.3.1 omni_broadcast

Variable	Description
central_module_c	int8 Possible values: 'B'.
server_type_c	int8 Possible values: 'N'.
broadcast_number_n	uint16 Possible values: 1.
network_status_x	omni_netwrksts See omni_netwrksts sub structure below.

3.2.3.2 omni_netwrksts

Variable	Description
status_u	uint32 Possible values: -208002 = OMNI_DU_DATALOST -208004 = OMNI_DU_LINKLOST -208006 = OMNI_DU_SENDERLOST -208008 = OMNI_DU_SENDERRESTARTED. The message text relating to the codes above can be retrieved through the omniapi_get_message_ex() function. For further explanation see the table below. Note: Negative numbers are being used here in an unsigned integer field. Participants need to cater for this situation until ASX Trade implements a new network message structure.
msglen_n	uint16 Indicates the size of the message array that immediately follows this structure.
	char[] The message text that immediately follows the structure, but it is not part of the structure. The length of the network status message is the size of omni_broadcast + the value in msglen_n.

3.2.3.3 Message Text

Status Value	Description
OMNI_DU_DATALOST	<p>Indicates that one or more broadcasts may have been lost. If the OMdu detects that a broadcast is missing, it will initiate retransmission and buffer subsequent broadcasts until the missing one is received. If there are not enough resources available at the gateway or the connecting application for the data to be sent, then a BN1 with OMNI_DU_DATALOST will be generated. The broadcasts are lost and should be recovered with the relevant queries. The string associated with the broadcast provides the name of the sender service at which the losses were detected.</p> <p>A common cause of OMNI_DU_DATALOST is when an application is not consuming the API buffer fast enough. This API buffer exists in virtual memory of the client application. The OMNI_API_BDX_BUFFER_SIZE environment variable can be used to increase the API buffer size which is the number of broadcasts that the buffer can hold. The default setting is 5500. The buffer size can be significantly increased if required. A setting of 300,000 has been used by clients.</p>
OMNI_DU_LINKLOST	<p>The connection from the gateway to the broadcast system has been lost. This is a serious problem. The OI application should cancel any current subscriptions and begin recovery processing.</p>
OMNI_DU_SENDERLOST	<p>The gateway has lost the connection to a server of a broadcast. The gateway will receive new broadcasts from that server when it becomes available again.</p> <p>This event can be ignored by OI applications since ASX Trade typically has two instances of each server process running. If a primary server process fails then the standby instance of the process will become active and start sending broadcasts.</p> <p>The event can be interpreted as 'fail-over in progress'.</p>

3.3 BN4 Network Status

3.3.1 Broadcast Function

The BN4 broadcast has the same structure as the BN1. This broadcast relates information on the participant's connection to ASX Trade. Currently in ASX Trade there is only one use for this broadcast – to tell the participants that they are about to be logged off. Participants can subscribe to these broadcasts using the event type of OMNI_EVTTYP_NETWORK (= 1001).

3.3.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	omni_broadcast
Information Type	Not Applicable

3.3.3 Message Structure

3.3.3.1 omni_broadcast

Variable	Description
central_module_c	int8

Variable	Description
	Possible values: 'B'.
server_type_c	int8 Possible values: 'N'.
broadcast_number_n	uint16 Possible values: 4.
network_status_x	omni_netwrksts See omni_netwrksts sub structure below.

3.3.3.2 omni_netwrksts

Variable	Description
status_u	uint32 Possible value: 4004 = OMNI_NE_LOGOUT_IMMINENT The above status indicates that the participant is about to be logged off. It is issued to those users who are still logged on when the system is about to run its nightly batch jobs. Other event numbers listed in the omni.h file can be ignored as they are not used in ASX Trade. The message text relating to the codes above can be retrieved through the omniapi_get_message_ex() function. Note: Negative numbers are being used here in an unsigned integer field. Participants need to cater for this situation until ASX Trade implements a new network message structure.
msglen_n	uint16 Indicates the size of the message array that immediately follows this structure.
	char[] The message text that immediately follows the structure, but is not part of the structure. The length of the network status message is the size of omni_broadcast + the value in msglen_n.

4 BD2 Edited Price Information

4.1 Broadcast Function

This broadcast provides information related to trade statistics. It is sent out when the Open, High, Low or Last price changes on a series or the traded volume. It is currently subjected to a 500 millisecond hold back. This means that the broadcast can contain the accumulated total of several trades occurring within that period.

The broadcast will also be sent when an index is changed. At regular intervals during the day, indices are calculated and the new values are disseminated.

In order to get a baseline of information for this, broadcast users are required to issue an IQ18 or IQ19 Total Volume and Prices query (refer to *ASX Trade Queries* for more information).

4.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead it holds a sequence of possible structs which are described below.
Information Type	Instrument Class
Virtual Underlying	True

4.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_hdr_t
- one or more sequences of:
 - item_hdr_t
 - one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - market_info_series_t (named structure 33038)
 - market_info_reason_t (named structure 33043)
 - market_info_base_t (named structure 33034)
 - market_info_asx_t (named structure 33039)
 - market_info_index_t (named structure 33040)
 - ob_levels_closing_t (named structure 33031)
 - market_info_asx_extended (named structure 33139)

4.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'D', 2}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t

Variable	Description
	The total size of the message, including this header.

4.3.2 item_hdr_t

Variable	Description
items_n	uint16_t The number of sub-items following this item header.
size_n	uint16_t The total size of the following sub-items, including this header.

4.3.3 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 The size of the structure that is to follow, including this header.

4.3.4 market_info_series_t (named structure 33038)

Variable	Description
series	series_t Contains the series (which could be an index) to which this price information relates.
reserved_i	int32_t Ignore. Reserved for future use.
all_or_none_c	uint8_t Ignore. Always 2.
filler_3_s	char[3] Ignore. Used for byte alignment.

4.3.5 market_info_reason_t (named structure 33043)

Variable	Description
edited_price_info_reason_c	uint8_t Ignore. Always 2.
filler_3_s	char[3] Ignore. Used for byte alignment.

4.3.6 market_info_base_t (named structure 33034)

This variable structure is provided in the broadcast only if any of the included fields has a new value. This named structure will not appear in the broadcast when the broadcast is sent due to an index change.

Variable	Description
opening_price_i	int32_t The first on-market sale price of the series for the day.

Variable	Description
	<p>If the 31st bit (highest bit) is set and the rest are zero, then this indicates that there is no opening price available. This differs from the value of zero (all bits zero) indicating a price of zero.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
high_price_i	<p>int32_t</p> <p>The highest on-market traded price of the series for the day.</p> <p>If the 31st bit (highest bit) is set and the rest are zero, then this indicates that there is no high price available. This differs from the value of zero (all bits zero) indicating a price of zero.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
low_price_i	<p>int32_t</p> <p>The lowest on-market traded price of the series for the day.</p> <p>If the 31st bit (highest bit) is set and the rest are zero, then this indicates that there is no low price available. This differs from the value of zero (all bits zero) indicating a price of zero.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
last_price_i	<p>int32_t</p> <p>The last on-market traded price of the series for the day.</p> <p>If the 31st bit (highest bit) is set and the rest are zero, then this indicates that there is no last price available. This differs from the value of zero (all bits zero) indicating a price of zero.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
volume_u	<p>int64_t</p> <p>Volume of the latest trade.</p>
turnover_u	<p>int64_t</p> <p>Total traded volume during the day.</p>
number_of_deals_u	<p>uint32_t</p> <p>Number of trades executed. This processed price information could be due to a number of orders matching within a very short period of time. The one broadcast could then possibly collate several trades.</p>
hhmmss_s	<p>char[6]</p> <p>The UTC time when this processed price information was issued. Given in HHMMSS format.</p>
trend_indicator_c	<p>char[1]</p> <p>Trend indicator for new price compared to previous one:</p> <p>'+' = higher price than previously</p> <p>'-' = lower price than previously</p> <p>'=' = same price as previously</p> <p>'' = no trend available.</p>
deal_source_c	<p>uint8_t</p> <p>Where the last trade was executed. Refer to <i>Trade Source</i> in <i>ASX Trade Introduction and Business Information</i> for the possible values in this field.</p>

4.3.7 market_info_asx_t (named structure 33039)

Variable	Description
number_of_trades_u	uint32_t Number of trades executed during the day.
turnover_value_q	int64_t The total traded amount today. Decimal places are implied from the attributes of the associated instrument class to the series.

4.3.8 market_info_index_t (named structure 33040)

This structure will appear in the broadcast when the broadcast is sent due to an index change.

Variable	Description
high_price_i	int32_t The highest value for an index during the day. If the 31 st bit (highest bit) is set and the rest are zero, then this indicates that there is no high value available. This differs from the value of zero (all bits zero) indicating an index value of zero. Decimal places are implied from the attributes of the associated instrument class to the series.
low_price_i	int32_t The lowest value during the day. If the 31 st bit (highest bit) is set and the rest are zero, then this indicates that there is no low value available. This differs from the value of zero (all bits zero) indicating an index value of zero. Decimal places are implied from the attributes of the associated instrument class to the series.
last_price_i	int32_t The last value given during the day. If the 31 st bit (highest bit) is set and the rest are zero, then this indicates that there is no last value available. This differs from the value of zero (all bits zero) indicating an index value of zero. Decimal places are implied from the attributes of the associated instrument class to the series.
change_previous_i	int32_t Change in percent since previous broadcast. Decimal places are implied from the attributes of the associated instrument class to the series.
change_yesterday_i	int32_t Percentage change since current day's initial value. Decimal places are implied from the attributes of the associated instrument class to the series.
points_of_movement_i	int32_t Points change since the current day's initial value. Decimal places are implied from the attributes of the associated instrument class to the series.

Variable	Description
date_time_of_dist_s	char[14] UTC date and time for distribution.
date_time_of_comp_s	char[14] UTC date and time for computation.

4.3.9 ob_levels_closing_t (named structure 33031)

This structure is provided in the broadcast only if any of the included fields have a new value.

Variable	Description
closing_price_i	int32_t This field contains the previous day's closing price. If the 31 st bit (highest bit) is set and the rest are zero, then this indicates that there is no close price available. This differs from the value of zero (all bits zero) indicating closing price of zero. Decimal places are implied from the attributes of the associated instrument class to the series.
open_balance_u	int64_t Indicates the open interest on the series, i.e. the number of outstanding contracts (this is not updated intraday).

4.3.10 market_info_asx_extended (named structure 33139)

Variable	Description
extended_opening_price_i	int64_t This field may be set to the opening price with up to four decimal places. If no extended opening price is received for the trade, the value defaults to the minimum value for the field type, INT64_T (-9223372036854775808)
extended_high_price_i	int64_t This field may be set to the high price with up to four decimal places. If no extended highest traded price is received for the trade, the value defaults to the minimum value for the field type, INT64_T (-9223372036854775808)
extended_low_price_i	int64_t This field may be set to the low price with up to four decimal places. If no extended lowest traded price is received for the trade, the value defaults to the minimum value for the field type, INT64_T (-9223372036854775808)
extended_last_price_i	int64_t This field may be set to the last price with up to four decimal places. If no extended last traded price is received for the trade, the value defaults to the minimum value for the field type, INT64_T (-9223372036854775808)

5 BI7 Information Ready

5.1 Broadcast Function

This broadcast is used to notify processes and applications that certain information is at hand, or that specific events have occurred.

Since this broadcast has an information type of “General” it is not possible for a user to filter their subscription, nor is it possible for ASX to filter the dissemination of the broadcast. Therefore if a participant, who is restricted to certain markets and instruments, subscribes to this broadcast, they will still receive the broadcasts that relate to markets and instruments that are unknown to them. This situation must be handled by the OI application.

5.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	info_ready_t
Information Type	General

5.3 Message Structure

5.3.1 info_ready_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 7}.
info_type_i	int32_t The type of information that is ready. Possible values: 100 = Settlement information 13 = All securities closed 1 = Trade information.
series	series_t Indicates for which market this broadcast was sent.
business_date_s	char[8] The business date for which the broadcast was sent. Format: YYYYMMDD.
sent_date_s	char[8] Actual date the broadcast was sent. Format: YYYYMMDD.
sent_time_s	char[6] Actual time the broadcast was sent. Format: HHMMSS.
clearing_date_s	char[8] The date of clearing in YYYYMMDD format.
seq_num_srm_n	uint16_t A unique sequence number.

6 BI41 Instrument Status Info

6.1 Broadcast Function

This broadcast provides information on the session state for a market, an instrument type, an instrument class, series, underlying or a linked underlying.

It is sent at the actual state change and can also be sent as a warning before the state changes. The “Seconds to State Change” field tells whether it is a warning or an actual state change. A value larger than zero in this field means that the broadcast is a warning.

A trading session state (TSS) is configurable on market level, instrument type level or instrument class level. An instrument session state (ISS) is configurable on instrument series level, underlying level or linked underlying level. In order to correctly calculate a particular instrument’s active session state users must follow the algorithm described in *ASX Trade Introduction and Business Information*.

The *UQ15 Instrument Status* query can be used to recover the information in this broadcast if the user missed it. Refer to *ASX Trade Queries* for more information.

Furthermore, since this broadcast has an information type of “General” it is not possible for a user to filter their subscription, nor is it possible for ASX to filter the dissemination of the broadcast. Therefore if a user, who is restricted to certain markets and instruments, subscribes to this broadcast, they will still receive the broadcasts that relate to markets and instruments that are unknown to them. This situation must be handled by the OI application.

6.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	instrument_status_info_t
Information Type	General

6.3 Message Structure

6.3.1 instrument_status_info_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 41}.
items_n	uint16_t Indicates how many items are held in the item array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	instrument_status_info_item_t[9] See instrument_status_info_item_t sub structure below.

6.3.2 instrument_status_info_item_t

Variable	Description												
series	<p>series_t</p> <p>Used to identify a specific market, instrument type, instrument class, series, underlying or linked underlying. The state level field below, indicates to which level this broadcast relates. The table below indicates what fields will be filled in at what levels:</p> <table> <tr> <td>If identifying a level of:</td><td>The following fields will be filled:</td></tr> <tr> <td>Market</td><td>country_c market_c</td></tr> <tr> <td>Instrument type</td><td>country_c market_c instrument_group_c</td></tr> <tr> <td>Instrument class</td><td>country_c market_c instrument_group_c commodity_n</td></tr> <tr> <td>Series</td><td>country_c market_c instrument_group_c modifier_c commodity_n expiration_date_n strike_price_i Note: expiration_date_n and strike_price_i can be zero for some instruments. Also modifier_c can be zero for any instrument.</td></tr> <tr> <td>Underlying, linked underlying</td><td>commodity_n</td></tr> </table>	If identifying a level of:	The following fields will be filled:	Market	country_c market_c	Instrument type	country_c market_c instrument_group_c	Instrument class	country_c market_c instrument_group_c commodity_n	Series	country_c market_c instrument_group_c modifier_c commodity_n expiration_date_n strike_price_i Note: expiration_date_n and strike_price_i can be zero for some instruments. Also modifier_c can be zero for any instrument.	Underlying, linked underlying	commodity_n
If identifying a level of:	The following fields will be filled:												
Market	country_c market_c												
Instrument type	country_c market_c instrument_group_c												
Instrument class	country_c market_c instrument_group_c commodity_n												
Series	country_c market_c instrument_group_c modifier_c commodity_n expiration_date_n strike_price_i Note: expiration_date_n and strike_price_i can be zero for some instruments. Also modifier_c can be zero for any instrument.												
Underlying, linked underlying	commodity_n												
seconds_to_state_change_n	<p>uint16_t</p> <p>This identifies how many seconds that are left until a change of state. If the value is larger than zero then this broadcast is a warning. If the value is zero it means that it is the actual state change.</p> <p>0 = State change.</p> <p>An integer = Warning.</p>												
state_number_n	<p>uint16_t</p> <p>The binary representation of the session state. Possible values can be fetched through the Trading State query (DQ29).</p> <p>Note that for an Instrument Session only, a state_number_n field of zero indicates that the current Instrument Session State for that particular series is to end (the timing depends on the seconds_to_state_change_n field). Trading Session States do not have an "End State" signal such as this.</p>												
warning_msg_s	<p>char[80]</p> <p>This is a warning message that will be shown at a trading state change.</p>												
state_level_e	<p>uint16_t</p> <p>Indicates the level that a state applies to. Possible values:</p>												

Variable	Description
	1 = Market 2 = Instrument type 3 = Instrument class 4 = Instrument series 5 = Underlying, linked underlying.
actual_start_date_s	char[8] Actual UTC start date, together with actual_start_time_s, for an instrument session state.
actual_start_time_s	char[6] Actual UTC start time, together with actual_start_date_s, for a session state.
next_planned_start_date_s	char[8] Planned UTC start date, together with next_planned_start_time_s, for the next planned state change.
next_planned_start_time_s	char[6] Planned UTC start time, together with next_planned_start_date_s, for the next planned state change.
filler_2_s	char[2] Ignore. Used for byte alignment.

7 BI63 Settlement Price Update

7.1 Broadcast Function

This broadcast disseminates intraday calculated settlement prices.

Currently, ASX Trade does not regularly broadcast the current trading day's settlement prices during the trading period of that day. However, if an error was noted in a previous day's settlement price, then the corrected settlement price will be broadcasted using this mechanism.

To download current values for the preliminary settlement prices, the Query Preliminary Settlement Prices Transaction (II17) is used. Refer to *ASX Trade Queries* for more information.

7.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	settle_price_update_t
Information Type	General

7.3 Message Structure

7.3.1 settle_price_update_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 63}.
items_n	uint16_t Identifies how many items are held in the item array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	settle_price_update_item_t[50] Array of items – maximum 50 items. See settle_price_update_item_t sub structure below.

7.3.2 settle_price_update_item_t

Variable	Description
series	series_t Used to identify the series for this item.
settle_price_i	int32_t The daily settlement price for the series.
settlement_date_s	char[8] Settlement date in YYYYMMDD format.
settlement_price_type_c	uint8_t Settlement price type. Possible values: 2 = Normal.
filler_3_s	char[3]

Variable	Description
	Ignore. Used for byte alignment.

8 BI73 Undo Information Ready

8.1 Broadcast Function

This broadcast is sent when the Undo Information Ready is triggered for a certain information type to retract information previously signalled as ready with the BI7 *Information Ready* broadcast message.

8.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	undo_info_ready_t
Information Type	General

8.3 Message Structure

8.3.1 undo_info_ready_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 73}.
info_type_i	int32_t The type of information that is undone. Possible values: 100 = Settlement information 13 = All securities closed 1 = Trade information.
series	series_t Used to identify a specific market that is to have the information ready signal undone.
business_date_s	char[8] Business date for which the broadcast was sent. Format: YYYYMMDD.
clearing_date_s	char[8] The date of clearing in YYYYMMDD format.
sent_date_s	char[8] Actual date the broadcast was sent. Format: YYYYMMDD.
sent_time_s	char[6] Actual time the broadcast was sent. Format: HHMMSS.
seq_num_srm_n	uint16_t A unique sequence number.

9 BI81 Market Announcement

9.1 Broadcast Function

This broadcast sends market messages and company announcements to all users. Market Control Messages are normally sent for a specific market (or all markets) whereas a Company Announcement Message is normally sent for a specific Underlying or Instrument Series.

9.1.1 Market Control Message

A Market Control Message is sent when ASX Trading Operations want to send a message. It is normally sent to a whole market, i.e. with Level set to Market (destination_level_c = 1) but it can sometimes be sent on Underlying or Series level.

This message will be sent with the message type set to Market Message (message_information_type_c = 2). It can be sent with two different priorities: normal and high. When the message is sent with normal priority (message_priority_c = 2) it is called a General Market Control Message and when it is sent with high priority (message_priority_c = 3) it is called an Important Market Control Message.

9.1.2 Company Announcement

A Company Announcement is sent when individual companies want to send information to the market, which means that these messages are typically sent with Level set to Underlying (destination_level_c = 2) or Series level (destination_level_c = 3).

This message will be sent with the message type set to Company Announcement (message_information_type_c = 1). It can be sent with two different priorities, normal and high. When the message is sent with normal priority (message_priority_c = 2) it is called a Company Announcement and when it is sent with high priority (message_priority_c = 3) it is called a Sensitive Company Announcement.

9.1.3 Static Line Message

A Static Line Message can, for example, be used to announce upgrades, reminders etc. This message will be sent with the message type set to Static Line (message_information_type_c = 3). It can be sent with two different priorities: high and critical. When the message is sent with high priority (message_priority_c = 3) it is called a Static Line Message and when it is sent with critical priority (message_priority_c = 4) it is called a Critical Static Line Message.

9.1.4 Notice Received Message

A Notice Received Message will typically be sent out prior to a Company Announcement. The message announces that a Company Announcement will be sent out soon for a particular company. At the time of sending out this Message, the trading will normally be halted in the instruments relating to the company.

This message will be sent with the message type set to Notice Received (message_information_type_c = 4). It can only be sent with one priority: high (message_priority_c = 3). It is called a Notice Received Message.

9.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs and their names are described below.
Information Type	General

9.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_hdr_t
- one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - message_core_info_t (named structure 35001)
 - message_information_t (named structure 35002)
 - destination_item_t (named structure 35003)
 - document_url_t (named structure 35004).

9.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 81}.
items_n	uint16_t The number of sub-items following this header.
size_n	uint16_t The total size of the message, including this header.

9.3.2 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 The size of the structure that is to follow, including this header.

9.3.3 message_core_info_t (named structure 35001)

Variable	Description
sequence_number_u	uint32_t The sequence number for this broadcast, starting at 1 each day.
message_information_type_c	uint8_t Identifies the kind of message sent. Possible values: 1 = Company announcement 2 = Market message 3 = Static Line 4 = Notice Received.
message_source_s	char[80] Source of the message, e.g. a linked exchange or ASX Trading Operations.
yyyymmdd_s	char[8] The date of the broadcast in YYYYMMDD format.
hhmmss_s	char[6]

Variable	Description
	The time of the broadcast in HHMMSS format.
message_priority_c	uint8_t The priority of the message. Possible values: 1 = Low priority 2 = Medium priority 3 = High priority 4 = Critical priority
message_header_s	char[80] The text of the message.
update_status_note_c	uint8_t Ignore. Currently not used.
filler_3_s	char[3] Ignore. Used for byte alignment.

9.3.4 message_information_t (named structure 35002)

Variable	Description
items_n	uint16_t The number of items in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	message_information_item_t[10] Array of items – maximum 10 items. See <i>message_information_item_t</i> sub structure below.

9.3.5 message_information_item_t

Variable	Description
text_line_s	char[80] One line of text information. It can be assumed that these lines of text in the array are in the appropriate order for the message. For company announcements this field contains the file name without the extension (.txt, .pdf) for the full company announcement disseminated by the ASX ComNews Service.

9.3.6 destination_item_t (named structure 35003)

Variable	Description
series	series_t Indicates the market, underlying or series to which this broadcast relates. Note that wildcards can be used for the market level that would indicate that the announcement is for the entire exchange.
destination_level_c	uint8_t Possible values: 1 = Market level 2 = Underlying level 3 = Series level.

Variable	Description
filler_3_s	char[3] Ignore. Used for byte alignment.

9.3.7 document_url_t (named structure 35004)

Variable	Description
items_c	uint8_t Indicates the number of characters in url_link_s below.
url_link_s	char[255] A url that can be used to direct the user to the full company announcement. The field is not space padded.

10 BI741 Instrument Status Broadcast

10.1 Broadcast Function

This broadcast provides information on the status of an instrument series.

It is sent at the actual session state change.

A trading session state is configurable on market level, instrument type level or instrument class level. An instrument session state is configurable on instrument series level, underlying level or linked underlying level.

The BI741 will send all session state changes at series level. This means for example that when a market moves into a new trading session state, BI741s will be sent for each instrument series traded in that market, with each broadcast being able to accommodate up to 80 series. This also means that only the active session state is disseminated for each series. For example if an instrument series moves into an instrument session state of SUSPEND, a BI741 will be sent for that series. If the market that this series is traded in then moves into the OPEN trading session state, no new BI741 message will be sent for that series because its active session state is still SUSPEND, since SUSPEND has a higher priority than OPEN.

The UQ15 Instrument Status query can be used to recover the information in this broadcast if the user missed it. To retrieve the definitions on actual trading states so as to determine their priority, users need to issue a DQ29 Trading State Query.

The BI741 broadcast has an information type of “General” it is therefore not possible for a user to filter their subscription, nor is it possible for ASX to filter the dissemination of the broadcast. Therefore if a user, who is restricted to certain markets and instruments, subscribes to this broadcast, they will still receive the broadcasts that relate to markets and instruments that are unknown to them.

When a new instrument series is created intraday, ASX Trade will disseminate a BI741 broadcast to notify users of the new series’ session state. There may be instances where users will receive this BI741 before the BU124/BU126, notifying them of the new series. The OI application has to handle this situation.

10.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	me_instrument_status_info_t
Information Type	General

10.3 Message Structure

10.3.1 me_instrument_status_info_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'I', 741}.
reserved	uint32_t Not used. Ignore.
time_spec	time_spec_t Timestamp of session state change. See time_spec_t sub structure.

Variable	Description
state_number_n	uint16_t The binary representation of the trading state. Possible values can be obtained through the DQ29 Trading State Query.
items_n	uint16_t Number of items in the following array.
item	me_instrument_status_info_item_t[80] See me_instrument_status_info_item_t sub structure.

10.3.2 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since the Epoch (1970-01-01 00:00:00 UTC).
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

10.3.3 me_instrument_status_info_item_t

Variable	Description
series	series_t The series affected by this session state transition.

11 BL51 Price Limit Update

11.1 Broadcast Function

This broadcast provides the Price Limit Reference Price and the actual Price Limits for instrument series.

11.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	limit_change_t
Information Type	General

11.3 Message Structure

11.3.1 limit_change_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'L', 51}.
items_n	uint16_t Number of items in the following array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	limit_change_item_t[50] See limit_change_item_t sub structure.

11.3.2 limit_change_item_t

Variable	Description
series	series_t Used to identify the instrument series for this item.
upper_limit_i	int32_t Price, High Limit. Decimal places are implied for the attributes if the associated instrument class to the series.
lower_limit_i	int32_t Price, Low Limit. Decimal places are implied for the attributes if the associated instrument class to the series.
reference_premium_i	int32_t Price Limits Reference Price for the instrument series. Decimal places are implied for the attributes if the associated instrument class to the series.

12 BO2 Order Book Changes

12.1 Broadcast Function

In ASX Trade, this broadcast is only disseminated for series in equity and interest rate type markets. It contains the changes in the order book. The broadcast describes the alteration made to an order and contains the changed data. Users should note that there is no order detail provided for derivative and warrant type markets. Only five levels of order depth (aggregated) are provided in the BO14 broadcast for derivatives and warrants (refer to *BO14/BO15 Order Book with Levels*).

To obtain a market picture of the order book, users should subscribe to this broadcast, then while temporarily storing any messages, execute an MQ7 query. On completion of the query, the stored broadcasts can be applied using the sequence number. For each series, if the broadcast has a higher sequence number than the one retrieved in the query, then the action from ob_command_c should be applied, otherwise, the broadcast can be ignored. Users need to cater for the special case where there are no orders in a series and the response from the MQ7 query returns no data. For full details on this situation and the recommended practice for getting a complete market picture refer to *MQ7 Total Order Book* in *ASX Trade Queries*.

ASX recommends that participants consider the impact of subscribing to this broadcast for all instrument classes. Intelligence may need to be applied at the OI application level so as to only subscribe to the broadcast for a particular series when it is absolutely necessary.

The broadcast will indicate an order book change to be either an add, delete or an amend. The type of change is indicated in the ob_command_c field.

12.1.1 Add

When ob_command_c is equal to 0:

- sequence_number_u is a consecutive number per series.
- quantity_difference_i is equal to the mp_quantity_i which is the quantity of the order.

12.1.2 Delete

When ob_command_c is equal to 1:

- The deleted order is identified by the position held in the order book (ob_position_u) and by the order number (order_number_u). Remaining fields contain redundant information.

12.1.3 Amend

When ob_command_c is equal to 2:

- The amended order is identified by the position held in the order book (ob_position_u) and by the order number (order_number_u). This means that the content that has changed has not affected the position of the order in the order book.
- Quantity difference is the difference between old and new quantity. Indicated when the mp_quantity_i field is changed (quantity_difference_i = new mp_quantity_i - old mp_quantity_i).
- The fields that follow contain the values of the order after the amendment has taken place regardless of which field has been changed.
- Increasing an order's quantity and/or changing its price may affect the order's position in the market. Even when the position is not changed, this will result in a deletion and a subsequent addition being broadcasted.


Note:

combo_mark_c has a non-zero value when the order is derived from a standard or Tailor Made Combination (TMC) (so-called bait order). For the business rules concerning combination orders, refer to *ASX Trade Introduction and Business Information*.

12.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	ob_changes_no_id_t
Information Type	Instrument Class

12.3 Message Structure

12.3.1 ob_changes_no_id_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'O', 2}.
changes	changes_t Refer to changes_t sub structure below.
order_number_u	quad_word The identifier of the order.
order_no_id	order_no_id_t Refer to order_no_id_t sub structure below.

12.3.2 changes_t

Variable	Description
sequence_number_u	uint32_t The sequence number for this broadcast. Sequence numbers start at one per series per day.
ob_position_u	uint32_t Position of the order in the order book, relative to the top of the market where 1 = Top.
quantity_difference_i	int64_t When the mp_quantity_i of an existing order is changed, the difference is stored here. For orders that are amended, this field will always be negative, indicating a decrease in quantity. An increase in quantity would cause a change in market position and therefore produce a "deleted order" broadcast followed by an "add order" broadcast.
ob_command_c	uint8_t Indicates the type of change. Possible values: 0 = Add 1 = Delete 2 = Amend.
change_reason_c	uint8_t Indicates the reason for the change. Possible values: 1 = Order deleted

Variable	Description
	3 = Trade 4 = Order inactivated 5 = Order amended 6 = Order added 8 = Order price changed 9 = Order deleted by central system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Central system deleted day order 21 = Inactivated by system due to Instrument Session change 23 = Inactivated due to Purge 24 = Inactivate day orders 26 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market 28 = Order transferred from one user to another 30 = Order reload at normal system start 31 = Order reload at intraday Market Place restart 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to trading 41 = Quote deleted due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 48 = Market-to-Limit Sweep order converted to Limit order 49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero 50 = Sweep order reloaded without MAQ and mid-tick attribute. 100 = Cancelled due to connection loss
combo_mark_c	uint8_t Indicates if an order is derived from a standard or TMC (so-called bait order). 0 = Not a bait order 254 = Bait order.
mbo_flags_c	uint8_t Used as an indicator in instances of an apparently locked market that all of the BO2 broadcasts associated with one transaction have been delivered. Possible values: 0 = More BO2(s) to come 2 = Last BO2 for associated series and more BO2(s) to come 3 = Last BO2 for matching operation.

12.3.3 order_no_id_t

Variable	Description
series	series_t This is the series of the order.
mp_quantity_i	int64_t Shown quantity of the order or zero for an undisclosed order.
premium_i	int32_t The price of the order.

Variable	Description
	Note that the price for a TMC order can be positive, zero or negative.
block_n	uint32_t Ignore. Always 1.
exch_order_type_n	uint16_t Exchange specific order types. Ignore any values returned that are not in the list below: 4 = Market Bid order (only entered by ASX Trading Operations) 8 = Price Stabilisation/Green Shoe Order 32 = Undisclosed (use order_type_c to determine order type). Note: Short sell will not be shown in this broadcast.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
filler_1_s	char[1] Ignore. Used for byte alignment.

13 BO5 Firm Order Book

13.1 Broadcast Function

All order related activities for a participant are disseminated via this directed broadcast, for example, when a user enters, changes or deletes an order or when an order is matched by another order. Thereby it is possible for each user to keep an internal order book for the whole participant. For order placement and amendment, the BO5 broadcast contains the private broker information (e.g. `customer_info_s` and `ex_client_s` fields). This information is not contained in the public order detail broadcast (BO2) or the public trade broadcast (CB16).

In order to match this broadcast with the public trade broadcast (CB16), users must take the value in the `order_number_u` field from the trade broadcast and match this with the order identifier in this broadcast.



Note:

Participants should **not** assume that when a trade occurs the BO5 will be disseminated before its associated CB15 and CB16.

In order to maintain the real-time order book from the BO5 information, the participant application can use the MQ8 or MQ92 query to download a baseline set of order information.

BO5 broadcasts that are missed, may be recovered for the current and previous trading days with the MQ151 query using the sequence numbers from the broadcast. These sequence numbers are unique per ASX Trade partition and per participant (refer to *MQ151 Query Order Broadcast* in *ASX Trade Queries* for more information).



Note:

A participant may have gaps in the BO5 sequence numbers if they are not authorised to see all instruments.

BO5 broadcasts are not sent for Market Maker quote transactions (MO36) for quote placement and quote amendment. However, should a quote in the market be traded, or when a quote is deleted, at that time only will a BO5 broadcast be sent to the users of the Market Maker's participant. The BO5 indicates that the quote has traded or was deleted and provides the private client info details (`ex_client_s`, `customer_info_s` etc.). In ASX Trade, a BO5 transaction response with only the actual change (not the input transaction) will be issued when a quote entered trades immediately on input.

For transactions that can submit either an absolute or a delta quantity such as MO3 or MO36, the BO5 always returns the resulting absolute quantity. This is a special case for this broadcast and users should refer to the preamble in `hv_alter_trans_t` for more information.

The data contained in these broadcasts may span more than one BO5 message. In this case the broadcast behaves the same way as a segmented query. The segment number is set to 1 for the first segment, 2 for the second segment, etc. The last segment is always set to zero. Thus, for single segment broadcasts, the segment number is zero.



Note:

In the case of a Fill and Kill order with residual quantity, two `order_change_combined_t` items are generated. The first part states the remaining quantity after matching while the second part indicates that the rest of the quantity is deleted. However in the scenario that the Fill and Kill order trades with a large number of orders resulting in more than one BO5 being broadcast, `order_change_combined_t` in the first BO5 states the remaining quantity from this BO5. In the second BO5, `order_change_separate_t` states the remaining quantity from this BO5. This continues until the final BO5 received and `order_change_separate_t` states the final remaining quantity and `order_change_combined_t` indicates that the rest of the quantity is deleted.


Note:

When an incoming order for a single instrument is matched against a large number of orders, the first BO5 segment includes an `order_change_combined_t` struct which provides the remaining quantity based on the trade details included in this BO5. Similarly the second BO5 normally includes `order_change_separate_t` struct providing the remaining quantity from this BO5. This continues until the final BO5 where the `order_change_separate_t` provides the final residual quantity.

The exception to this behavior is when a combination order matches against a large number of outright orders in a leg(s) resulting in multiple BO5 segments. In this case an `order_change_separate_t` struct is only provided in the final BO5 because there is more than one single binary match involved and it is not possible to correctly provide the remaining quantity (via `order_change_separate_t`) for the combination order for each BO5 segment.

For example a combination order matched against one order in Leg1 and 100 orders in Leg2 would result in multiple BO5s. In terms of trade information there would be one leg trade match in Leg1 and then 100 for Leg2. The segmentation will occur somewhere within those 100 matches in Leg2. However at this point there are no correct values to provide the remaining combination order quantity (based on Leg1 a quantity of 100 of the combination order has traded, but based on the Leg2 less than 100 has traded).

In cases where the exception applies, an application needs to treat segmented BO5s as one. This can be achieved by utilising the `segment_number_n` included in each BO5 which indicates there are multiple BO5 segments involved, starting at 1 for the first segment, 2 for the second, and so on with a value of 0 indicating the last segment. For single segmented BO5 broadcasts, the value in the field is 0.

Logically the following pseudo code would apply:

If `segment_number_n > 0` and missing `order_change_separate_t` then concatenate segments.

13.2 Broadcast Properties

Function Call	<code>omniapi_read_event_ext_ex</code> or <code>omniapi_read_event_block</code>
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.
Information Type	Instrument Dedicated
Segmented	True

13.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- `broadcast_hdr_t`
- one or more sequences of:
 - `sub_item_hdr_t`
 - a choice of:
 - `block_price_trans_t` (named structure 34007)
 - `hv_alter_trans_t` (named structure 34010)
 - `hv_alter_trans_p_t` (named structure 34110)
 - `hv_order_trans_t` (named structure 34005)
 - `hv_order_trans_p_t` (named structure 34105)

- hv_price_2_trans_t (named structure 34001)
- multi_order_response_t (named structure 34906)
- order_change_combined_t (named structure 34902)
- order_change_separate_t (named structure 34903)
- order_chg_sep_trans_ack_t (named structure 34919)
- order_info_t (named structure 34917)
- order_price_change_t (named structure 34905)
- order_return_info_t (named structure 34904)
- order_trade_info_t (named structure 34920)
- order_leg_trade_info_t (named structure 34921)
- order_trade_info_asx_t (named structure 34922)
- segment_instance_number_t (named structure 34901)
- trade_report_1_trans_t (named structure 34021)
- trade_report_1_trans_p_t (named structure 34119)
- trade_report_2_trans_t (named structure 34022)
- exchange_info_t (named structure 50004)
- free_text_t (named structure 34801)
- clearing_info_t (named structure 34802)
- single_order_insert_t (named structure 34808)
- single_order_update_t (named structure 34809)
- basic_order_t (named structure 34810)
- reserve_order_t (named structure 34812)
- basic_order_update_t (named structure 34815)
- centre_point_order_t (named structure 34816)
- enhanced_cp_matching_t (named structure 34831)
- inactive_order_t (named structure 34818)
- order_submitter_t (named structure 34819)
- order_owner_t (named structure 34804)
- ranking_time_t (named structure 34949)
- crossing_t (named structure 34820)
- regulatory_t (named structure 34821)
- short_sell_order_t (named structure 34829)
- short_sell_order_change_t (named structure 34830).



Note:

The “_p” versions of the structures are sent when the transaction is entered by ASX Trading Operations on behalf of a participant. These structures need to be handled by the OI application.

The combinations of structures received for some of the common actions are given below:

- Enter MO4 Delete Order:
 - segment_instance_number_t (named structure 34901)
 - order_change_separate_t (named structure 34903)
- Enter MO1 Order (if order is unmatched):
 - segment_instance_number_t (named structure 34901)
 - single_order_insert_t (named structure 34808)
 - basic_order_t (named structure 34810)
 - clearing_info_t (named structure 34802)
 - free_text_t (named structure 34801)
 - exchange_info_t (named structure 50004)
 - regulatory_t (named structure 34821)

- order_return_info_t (named structure 34904)
- Enter MO1 Order (if order wholly or partially trades):
 - BO5 sent to the originator of the MO1 contains:
 - segment_instance_number_t (named structure 34901)
 - single_order_insert_t (named structure 34808)
 - basic_order_t (named structure 34810)
 - clearing_info_t (named structure 34802)
 - free_text_t (named structure 34801)
 - exchange_info_t (named structure 50004)
 - regulatory_t (named structure 34821)
 - order_return_info_t (named structure 34904)
 - order_return_info_t (named structure 34904)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_combined_t (named structure 34902)
 - BO5 sent to the other participant of the trade contains:
 - segment_instance_number_t (named structure 34901)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_separate_t (named structure 34903)
- Enter MO3 Update Single Order:
 - segment_instance_number_t (named structure 34901)
 - single_order_update_t (named structure 34809)
 - basic_order_update_t (named structure 34815)
 - order_return_info_t (named structure 34904)
- Enter MO75 One-sided Trade Report:
 - if the trade report is unmatched:
 - segment_instance_number_t (named structure 34901)
 - trade_report_1_trans_t (named structure 34021)
 - order_return_info_t (named structure 34904)
 - if the trade report is matched, the sender receives:
 - segment_instance_number_t (named structure 34901)
 - trade_report_1_trans_t (named structure 34021)
 - order_return_info_t (named structure 34904)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_combined_t (named structure 34902)
 - if the trade report is matched, the counterparty receives:
 - segment_instance_number_t (named structure 34901)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_separate_t (named structure 34903)
- Enter MO76 Two-side Trade Report:

- segment_instance_number_t (named structure 34901)
 - trade_report_2_trans_t (named structure 34022)
 - order_return_info_t (named structure 34904)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_combined_t (named structure 34902)
- Enter MO77 Crossing Combination Trade Report:
 - One BO5 per leg:
 - segment_instance_number_t (named structure 34901)
 - trade_report_2_trans_t (named structure 34022)
 - order_return_info_t (named structure 34904)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_trade_info_t (named structure 34920)
 - order_trade_info_asx_t (named structure 34922)
 - order_change_combined_t (named structure 34902).

13.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'O', 5}.
items_n	uint16_t The number of sub-items following this header.
size_n	uint16_t The total size of the message, including this header.

13.3.2 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 The size of the structure that is to follow, including this header.

13.3.3 block_price_trans_t (named structure 34007)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t The series to which this order information relates.
give_up_member	give_up_member_t Indicates the clearing identifier used for this order. See give_up_member_t sub structure.

Variable	Description
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 the participant their own order identifier.
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.
items_c	uint8_t The number of items in the array.
item	block_price_trans_item_t[14] Array of items – maximum 14 items. See block_price_trans_item_t sub structure.

13.3.3.1 block_price_trans_item_t

Variable	Description
series	series_t The series to which this order information relates.
order_number_bid_u	quad_word Indicates the identifier of the order on the bid side of this item in a block transaction.
order_number_ask_u	quad_word Indicates the identifier of the order on the ask side of this item in a block transaction.
bid_premium_i	int32_t Indicates the price of the order on the bid side of this item in a block transaction. Note: it is possible for a price to be 0, indicating a market order, or even negative for a combination.
ask_premium_i	int32_t Indicates the price of the order on the ask side of this item in a block transaction. Note that it is possible for a price to be zero, indicating a market order, or even negative for a combination.
bid_quantity_i	int64_t Indicates the shown quantity of the order on the bid side of this item in a block transaction.
ask_quantity_i	int64_t Indicates the shown quantity of the order on the ask side of this item in a block transaction.
bid_total_volume_i	int64_t Indicates the total quantity including any hidden quantity of the order on the bid side of this item in a block transaction.
ask_total_volume_i	int64_t Indicates the total quantity including any hidden quantity of the order 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)

Variable	Description
	1 = All other orders types.
time_validity_n	<p>uint16_t</p> <p>The time validity that applies to both orders in this item - on the bid and the ask side. This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</p> <p>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.</p> <p>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</p> <p>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.</p> <p>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.</p> <p>Note: The BO5 broadcast 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.</p> <p>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.</p> <p>Note: The system on order reload can set bits to indicate an updated order status. Bit 8(the most significant bit) in the unit byte is used to indicate a negative value in the number of units byte. A negative value in the number of unit’s byte is used to indicate the past, while a positive value is used to indicate the future. Bit 7 in the unit byte is used to indicate that the order has been inactivated due to a purge. Example Unit = 193 Count = 0 (i.e. binary 11000001 00000000, dec 49408). Rest of day order inactivated due to purge.</p>
order_type_c	<p>uint8_t</p> <p>Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values:</p> <p>1 = Limit price order (premium_i = an integer)</p> <p>2 = Market order (premium_i = 0)</p> <p>3 = Market-to-Limit order (premium_i = 0)</p> <p>17 = Best-Limit order (premium_i = 0, time_validity_n != 0).</p> <p>Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Centre Point, Centre Point Block and Sweep orders.</p>
ex_client_s	<p>char[10]</p> <p>Client – a free text field typically used to indicate to the participant the ultimate client making the order.</p>
delta_quantity_c	<p>uint8_t</p> <p>Usually, on an order amendment, this field indicates if the quantity values are the new values or the changed values supplied for each order. However, for any transactions that can submit either an absolute or a delta quantity, the BO5 will return only the resulting absolute quantity. Possible values:</p> <p>1 = Absolute quantity.</p>
filler_2_s	<p>char[2]</p> <p>Ignore. Used for byte alignment.</p>

13.3.4 hv_alter_trans_t (named structure 34010)

When an order is amended using a transaction that involves the hv_alter_trans_t structure, the corresponding BO5 does **not** return the values that were placed in the original transaction. This is different to other transactions whereby the exact same values given to ASX Trade are returned to the user.

All fields for the hv_alter_trans_t structure and its corresponding order_var_t structure will contain the resultant values on the order, except for give_up_member_t and exchange_info_s. The latter two fields are only populated if the user has sent changes to the order for those fields in the corresponding transaction. Therefore, if the values in the give_up_member_t struct and the exchange_info_s are set to NULLs, then it indicates that nothing has been changed for those values for this particular amendment. If they do however contain values, then just like all the other fields it indicates their present values. The OI application must handle this situation.

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t This is the series of the order.
order_number_u	quad_word The identifier of the order that is being amended.
order_var	order_var_t See order_var_t sub structure order_var_t for more information.
give_up_member	give_up_member_t See give_up_member_t sub structure.
exchange_info_s	char[32] A free text field used at the participant's discretion.
total_volume_i	int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as > 0 for iceberg orders. For normal orders this is set as zero. normal order: total_volume_i = 0, display_quantity_i != 0 iceberg order: total_volume_i = display_quantity_i != 0 iceberg order: total_volume_i > display_quantity_i != 0.
delta_quantity_c	uint8_t Usually, on an order amendment, this field indicates if the quantity values are the new values or the changed values supplied for each order. However, for any transactions that can submit either an absolute or a delta quantity, the BO5 will return only the resulting absolute quantity. Possible value: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.
balance_quantity_i	int64_t Indicates that the participant wants a check done on the quantity of the existing order before amendment. Possible values: 0 = No balance check is performed Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.

13.3.5 hv_alter_trans_p_t (named structure 34110)

As per hv_alter_trans_t, this structure returns the resultant values of an amended order, with some exceptions. Refer to the explanation for hv_alter_trans_t.

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t This is the series of the order.
trading_code	trading_code_t See trading_code_t sub structure.
order_number_u	quad_word The identifier of the order that is being amended.
order_var	order_var_t See order_var_t sub structure for more information.
give_up_member	give_up_member_t See give_up_member_t sub structure for more information.
exchange_info_s	char[32] A free text field used at the participant's discretion.
total_volume_i	int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) volume. Only set as > 0 for iceberg orders. For normal orders this is set as 0. normal order: total_volume_i = 0, display_quantity_i != 0. iceberg order: total_volume_i = display_quantity_i != 0. iceberg order: total_volume_i > display_quantity_i != 0.
delta_quantity_c	uint8_t Usually, on an order amendment, this field indicates if the quantity values are the new values or the changed values supplied for each order. However, for any transactions that can submit either an absolute or a delta quantity, the BO5 will return only the resulting absolute quantity. Possible value: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.
balance_quantity_i	int64_t Indicates that the user wants a check done on the quantity of the existing order before amendment. Possible value: 0 = No balance check is performed. Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.

13.3.6 hv_order_trans_t (named structure 34005)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.

Variable	Description
series	series_t The series to which this order information relates.
order_var	order_var_t See <i>order_var_t</i> sub structure for more information.
give_up_member	give_up_member_t See <i>give_up_member_t</i> sub structure.
exchange_info_s	char[32] A free text field used at the participant's discretion.
total_volume_i	int64_t The total quantity of the order i.e. both the hidden and shown (mp_quantity_i in order_var_t) quantity. Only set as > 0 for iceberg orders. For normal orders this is set as 0. normal order: total_volume_i = 0, display_quantity_i != 0. iceberg order: total_volume_i = display_quantity_i != 0. iceberg order: total_volume_i > display_quantity_i != 0.

13.3.7 hv_order_trans_p_t (named structure 34105)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t The series to which this order information relates.
trading_code	trading_code_t See <i>trading_code_t</i> sub structure.
order_var	order_var_t See <i>order_var_t</i> sub structure.
give_up_member	give_up_member_t See <i>give_up_member_t</i> sub structure.
exchange_info_s	char[32] A free text field used at the participant's discretion.
total_volume_i	int64_t The total quantity of the order i.e. both the hidden and shown (mp_quantity_i in order_var_t) quantity. Only set as > 0 for iceberg orders. For normal orders this is set as 0. normal order: total_volume_i = 0, display_quantity_i != 0. iceberg order: total_volume_i = display_quantity_i != 0. iceberg order: total_volume_i > display_quantity_i != 0.

13.3.8 hv_price_2_trans_t (named structure 34001)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t

Variable	Description
	The series to which this order information relates.
give_up_member	give_up_member_t See <i>give_up_member_t</i> sub structure.
order_number_bid_u	quad_word Indicates the identifier of the order on the bid side of this block transaction.
order_number_ask_u	quad_word Indicates the identifier of the order on the ask side of this block transaction.
bid_premium_i	int32_t Indicates the price of the order on the bid side of this block transaction. Note: it is possible for a price to be 0, indicating a market order, or even negative for a combination.
ask_premium_i	int32_t Indicates the price of the order on the ask side of this block transaction. Note: it is possible for a price to be 0, indicating a market order, or even negative for a combination.
bid_quantity_i	int64_t Indicates the shown quantity of the order on the bid side of this block transaction.
ask_quantity_i	int64_t Indicates the shown quantity of the order on the ask side of this block transaction.
bid_total_volume_i	int64_t Indicates the total quantity of the order on the bid side of this block transaction.
ask_total_volume_i	int64_t Indicates the total quantity of the order on the ask side of this 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 orders in this block transaction - on the bid and the ask side. 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 today. Expiry will occur at the end of day’s trading on final day. Note: The BO5 broadcast 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 Cancel” type. Order will be valid for the maximum allowed time for that particular instrument type.

Variable	Description
ex_client_s	char[10] Client – a free text field typically used to indicate to ASX 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) 2 = Market order (premium_i = 0) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0, 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, Centre Point, Centre Point Block and Sweep orders.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.
exchange_info_s	char[32] A free text field used at ASX participant's discretion. It is not used or visible to participants.
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.

13.3.9 multi_order_response_t (named structure 34906)

The multi_order_response_t structure is sent in a BO5 originating from a received block quote (MO36). It contains information about failed orders of the block quote.

Variable	Description
transaction_status_i	int32_t Contains the same information as returned from the omniapi_tx_ex function, indicating the status. Possible values: 0 = Successful. Any other value = Transaction aborted.
trans_ack_i	int32_t Contains the same information as the transaction status from the omniapi_tx_ex function. The codes vary depending on the context in which they occur, but some common examples would be: 1 = No part of the order placed in the order book and no part closed 2 = The whole order closed 3 = The order partially closed and nothing placed in the order book 4 = The whole order placed in the order book 6 = The order partially placed in the order book and partially closed. GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type. MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.
item_number_c	uint8_t The item number in a quote transaction with a value of zero representing the first item.

Variable	Description
	Note: This does not indicate which side of the quote failed. It could be either the bid or the ask or both.
filler_3_s	char[3] Ignore. Used for byte alignment.

13.3.10 order_change_combined_t (named structure 34902)

When an order entered into the system is modified (such as traded) in any way before being added to the order book, an *order_change_combined_t* structure is sent in the same broadcast.



Note:

In the case of a Fill and Kill order with residual quantity, two *order_change_combined_t* items are generated. The first part states the remaining quantity after matching while the second part indicates that the residual quantity is deleted. However in the scenario that the Fill and Kill order trades with a large number of orders resulting in more than one BO5 being broadcast, *order_change_combined_t* in the first BO5 states the remaining quantity from this BO5. In the second BO5, *order_change_separate_t* states the remaining quantity from this BO5. This continues until the final BO5 is received when *order_change_separate_t* states the final residual quantity and *order_change_combined_t* indicates that the rest of the quantity is deleted (i.e. killed).



Note:

When an incoming order for a single instrument is matched against a large number of orders, the first BO5 segment includes an *order_change_combined_t* struct which provides the remaining quantity based on the trade details included in this BO5. Similarly the second BO5 normally includes *order_change_separate_t* struct providing the remaining quantity from this BO5. This continues until the final BO5 where the *order_change_separate_t* provides the final residual quantity.

The exception to this behaviour is when a combination order matches against a large number of outright orders in a leg(s) resulting in multiple BO5 segments. In this case an *order_change_separate_t* struct is only provided in the final BO5 because there is more than one single binary match involved and it is not possible to correctly provide the remaining quantity (via *order_change_separate_t*) for the combination order for each BO5 segment.

For example a combination order matched against one order in Leg1 and 100 orders in Leg2 would result in multiple BO5s. In terms of trade information there would be one leg trade match in Leg1 and then 100 for Leg2. The segmentation will occur somewhere within those 100 matches in Leg2. However at this point there are no correct values to provide the remaining combination order quantity (based on Leg1 a quantity of 100 of the combination order has traded, but based on the Leg2 less than 100 has traded).

In cases where the exception applies, an application needs to treat segmented BO5s as one. This can be achieved by utilising the *segment_number_n* included in each BO5 which indicates there are multiple BO5 segments involved, starting at 1 for the first segment, 2 for the second, and so on with a value of 0 indicating the last segment. For single segmented BO5 broadcasts, the value in the field is 0.

Logically the following pseudo code would apply:

If *segment_number_n* > 0 and missing *order_change_separate_t* then concatenate segments.

Variable	Description
mp_quantity_i	int64_t Indicates the remaining quantity of the order, after this change was done. A zero indicates the whole order was deleted.
total_volume_i	int64_t

Variable	Description
	This is the total remaining volume for iceberg orders. It includes the hidden and shown volumes.
item_number_c	uint8_t The item number of a block transaction that relates to this change.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
change_reason_c	uint8_t Indicates the reason for the change. Possible values include: 1 = Order deleted 3 = Trade 4 = Order inactivated 5 = Order amended 6 = Order added 8 = Order price changed 9 = Order deleted by central system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Central system deleted day order 21 = Inactivated by system due to Instrument Session change 23 = Inactivated due to Purge 24 = Inactivate day orders 26 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market 28 = Order transferred from one user to another 30 = Order reload at normal system start 31 = Order reload at intraday Market Place restart 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to trading 41 = Quote deleted due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 48 = Market-to-Limit Sweep order converted to Limit order 49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero 50 = Sweep order reloaded without MAQ and mid-tick attribute.
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.11 order_change_separate_t (named structure 34903)

The *order_change_separate_t* structure is sent out due to changes in quantity of orders residing in the order book. As with *order_change_combined_t*, the quantity and total volume fields describe the remaining volumes of the order.

Variable	Description
series	series_t The series to which this order information relates.

Variable	Description
order_number_u	quad_word Indicates the identifier of the order.
mp_quantity_i	int64_t Indicates the remaining quantity of the order, after this change was done. A zero indicates the whole order was deleted.
total_volume_i	int64_t This is the total remaining volume for iceberg orders. It includes the hidden and shown volumes.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
change_reason_c	uint8_t Indicates the reason for the change. Possible values: 1 = Order deleted 3 = Trade 4 = Order inactivated 5 = Order amended 6 = Order added 8 = Order price changed 9 = Order deleted by central system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Central system deleted day order 21 = Inactivated by system due to Instrument Session change 23 = Inactivated due to Purge 24 = Inactivate day orders 26 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market 28 = Order transferred from one user to another 30 = Order reload at normal system start 31 = Order reload at intraday Market Place restart 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to trading 41 = Quote deleted due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 48 = Market-to-Limit Sweep order converted to Limit order 49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero 50 = Sweep order reloaded without MAQ and mid-tick attribute.
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.

Variable	Description
filler_1_s	char[1] Ignore. Used for byte alignment.
originator_trading_code	originator_trading_code_t Indicates the user that sent the transaction that caused this change. See originator_trading_code_t sub structure.
execution_timestamp	time_spec_t See time_spec_t sub structure.

13.3.12 order_chg_sep_trans_ack_t (named structure 34919)

Variable	Description
trans_ack_i	int32_t Contains the same information as the transaction status from the omniapi_tx_ex function. The codes vary depending on the context in which they occur, but some common examples would be: 1 = No part of the order placed in the order book and no part closed 2 = The whole order closed 3 = The order partially closed and nothing placed in the order book 4 = The whole order placed in the order book 6 = The order partially placed in the order book and partially closed. GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type. MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.
order_change_separate	order_change_separate_t See order_change_separate_t (named structure 34903).

13.3.13 order_info_t (named structure 34917)

Variable	Description
timestamp_in	time_spec_t See time_spec_t sub structure.
timestamp_created	time_spec_t See time_spec_t sub structure.
order_number_u	quad_word Indicates the identifier of the order.
party	party_t See party_t sub structure below.
order	order_t See order_t sub structure below.
total_volume_i	int64_t Current total quantity of the order.
display_quantity_i	int64_t Current shown quantity of the order.
orig_total_volume_i	int64_t The total quantity of the order when it was originally entered.

Variable	Description
orig_shown_quantity_i	int64_t The shown quantity of the order when it was originally entered.
order_state_u	uint32_t Indicates the state current state of the order. Possible values: 10 = Active order 11 = Central inactive order.

13.3.14 order_t

Variable	Description
series	series_t The series to which this information relates.
trading_code	trading_code_t Indicates the user who originally placed the order. See trading_code_t sub structure.
order_var	order_var_t See order_var_t sub structure.
ex_user_code	ex_user_code_t Indicates the user who last changed the order. See ex_user_code_t sub structure.
give_up_member	give_up_member_t Indicates the clearing participant. See give_up_member_t sub structure.
exchange_info_s	char[32] A free text field used at the participant's discretion.
order_index_u	uint32_t Ignore. Currently not used.
transaction_number_n	uint16_t The transaction number that was used to originally enter this order (e.g. 1 indicates MO1).
change_reason_c	uint8_t 1 = Order deleted 3 = Trade 4 = Order inactivated 5 = Order amended 6 = Order added 8 = Order price changed 9 = Order deleted by central system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Central system deleted day order 21 = Inactivated by system due to Instrument Session change 23 = Inactivated due to Purge 24 = Inactivate day orders 26 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market 28 = Order transferred from one user to another

Variable	Description
	30 = Order reload at normal system start 31 = Order reload at intraday Market Place restart 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to trading 41 = Quote deleted due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 48 = Market-to-Limit Sweep order converted to Limit order 49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero 50 = Sweep order reloaded without MAQ and mid-tick attribute.
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.15 order_price_change_t (named structure 34905)

The order_price_change structure is sent out for orders for which the price has been changed.

Variable	Description
series	series_t The series to which this order information relates.
order_number_u	quad_word Indicates the identifier of the order.
premium_i	int32_t Indicates the price of the order.
execution_timestamp	time_spec_t See time_spec_t sub structure.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
change_reason_c	uint8_t 1 = Order deleted 3 = Trade 4 = Order inactivated 5 = Order amended 6 = Order added 8 = Order price changed 9 = Order deleted by central system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Central system deleted day order 21 = Inactivated by system due to Instrument Session change 23 = Inactivated due to Purge 24 = Inactivate day orders 26 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market

Variable	Description
	28 = Order transferred from one user to another 30 = Order reload at normal system start 31 = Order reload at intraday Market Place restart 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to trading 41 = Quote deleted due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 48 = Market-to-Limit Sweep order converted to Limit order 49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero 50 = Sweep order reloaded without MAQ and mid-tick attribute.
filler_2_s	char[2] Ignore. Used for byte alignment.

13.3.16 order_return_info_t (named structure 34904)

The *order_return_info_t* structure indicates the return status that the participant received after entering the transaction.

Variable	Description
trans_ack_i	int32_t The answer to the participant. Contains the transaction status returned in the omniapi_tx_ex function. Return codes vary depending on the context in which they occur, but some common examples would be: 1 = No part of the order placed in the order book and no part closed 2 = The whole order closed 3 = The order partially closed and nothing placed in the order book 4 = The whole order placed in the order book 6 = The order partially placed in the order book and partially closed. GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type. MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.
order_number_u	quad_word Indicates the identifier of the order that was returned to the user.
originator_trading_code	originator_trading_code_t Indicates the user that sent the transaction that caused this change. See originator_trading_code_t sub structure.
execution_timestamp	time_spec_t See time_spec_t sub structure.

13.3.17 order_trade_info_t (named structure 34920)

Variable	Description
match_id	match_id_t

Variable	Description
	Match ID represents a unique combination of execution event number, execution group number and match item number.
trade_price_i	int32_t Defines the trade price.
trade_quantity_i	int64_t Defines the trade quantity.
item_number_c	uint8_t The item number in a quote transaction.
deal_source_c	uint8_t Where the trade is created. Refer to <i>Trade Source</i> in <i>ASX Trade Introduction and Business Information</i> for the possible values in this field.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.18 order_leg_trade_info_t (named structure 34921)

Variable	Description
series	series_t Contains the series to which this trade information relates.
match_id	match_id_t Match ID represents a unique combination of execution event number, execution group number and match item number.
order_number_u	quad_word Identifies the order taking part in this side of the trade.
trade_price_i	int32_t Defines the trade price.
trade_quantity_i	int64_t Defines the trade quantity.
item_number_c	uint8_t The item number in a quote transaction.
deal_source_c	uint8_t Where the trade is created. Refer to <i>Trade Source</i> in <i>ASX Trade Introduction and Business Information</i> for the possible values in this field.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.19 order_trade_info_asx_t (named structure 34922)

Variable	Description
opposing_order_number_u	quad_word Order number for the opposing order for this trade.
trade_condition_n	uint16_t The condition in which a trade was executed. Possible values: 0 = No Condition 2 = Internal Trade/Crossing 8 = Buy Write (Equity/Derivative Combination). This field acts as a bit mask. The binary AND operator can be used on the above possible values. Refer to ASX Trade Markets, Instrument Groups and Trade Condition Codes .
exch_order_type_n	uint16_t Exchange specific order types. Ignore any values returned that are not in the list below: 2 = Short Sell order. Can also be combined with other exchange specific order types/attributes) outlined below: 4 = Market Bid order 8 = Price Stabilisation/Green Shoe Order 32 = Undisclosed 64 = Centre Point Order 2048 = Sweep order 2050 = Short Sell Sweep Order 4096 = Centre Point Block order.
ext_t_state_c	uint8_t Trade report code. Possible values: 0 = Ignore, not relevant. See <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> for more information.
opposing_deal_source_c	uint8_t Deal Source for the opposing side of this trade.
aggressive_c	uint8_t An aggressive order is the order that immediately results in a trade. A passive order does not result in an immediate trade. Passive (Price taker) = 0 (zero) Aggressive (Price maker) = 1.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Bid 2 = Ask.
trade_short_sell_quantity_i	int64_t Traded short sell quantity, i.e. the portion of the trade that was short.
counter_order_capacity_c	char[1] Dealing capacity of the counter order for crossings. For crossed trades, this returns the value provided in capacity_of_participant_s of the opposing order. For non- crossed trades, zero will be returned.
filler_3_s	filler_3_s Ignore. Used for byte alignment.

13.3.20 segment_instance_number_t (named structure 34901)

It is possible for the one BO5 broadcast to be split into several segments. The maximum size of a broadcast is approximately 14 KB.

Variable	Description
segment_number_n	uint16_t A BO5 broadcast can be segmented if the one broadcast contains too much information to be sent over the OMdu at once. This field will indicate the number of the segment of the BO5, starting at 1 for the first segment, 2 for the second, and so on with a value of 0 indicating the last segment. For single segmented BO5 broadcasts, the value in this field is 0.
instance_c	uint8_t Indicates from which partition this broadcast originated. A value of zero is given if only one partition exists.
filler_1_s	char[1] Ignore. Used for byte alignment.
sequence_number_u	uint32_t Sequence number assigned to the BO5 broadcast. When a BO5 broadcast is extended over several segments, the value in this field will still be incremented by one for each segment.
trading_code	trading_code_t The user who originally placed the order. See trading_code_t sub struct.

13.3.21 trade_report_1_trans_t (named structure 34021)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t This is the series of the order.
order_var	order_var_t See order_var_t sub structure below.
party	party_t This is the declared counterparty for the other side of the trade. See party_t sub structure.
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 See give_up_member_t sub structure.
settlement_date_s	char[8] The settlement date for the trade. Format: YYYYMMDD.
time_of_agreement_date_s	char[8]

Variable	Description
	The time when the trade was agreed, date part. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] The time when the trade was agreed, time part. Format: HHMMSS.
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.
short_sell_quantity_i	int64_t Quantity of the trade that is short (partial or whole). 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 0.

13.3.22 trade_report_1_trans_p_t (named structure 34119)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t This is the series of the order.
trading_code	trading_code_t The identity of the user who caused the BO5 to be sent. See trading_code_t sub structure.
order_var	order_var_t See order_var_t sub structure below.
party	party_t This is the declared counter party for the other side of the trade. See party_t sub structure.
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_exchang_info_t sub structure.
give_up_member	give_up_member_t See give_up_member_t sub structure.
settlement_date_s	char[8] The settlement date for the trade. Format: YYYYMMDD.
time_of_agreement_date_s	char[8] The time when the trade was agreed, date part. Format: YYYYMMDD.

Variable	Description
time_of_agreement_time_s	char[6] The time when the trade was agreed, time part. Format: HHMMSS.
deferred_publication_c	uint8_ 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.
short_sell_quantity_i	int64_t Quantity of the trade that is short (partial or whole). 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.

13.3.23 trade_report_2_trans_t (named structure 34022)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t The series for which the trade report exists.
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.
time_of_agreement_date_s	char[8] The time when the trade was agreed, date part. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] The time when the trade was agreed, time part. Format: HHMMSS.
ext_t_state_c	uint8_t Trade report code. Possible values: 0 = Ignore, not relevant. See <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> .
deferred_publication_c	uint8_t Ignore. Not used by ASX Trade.
bid_side	trd_rpt_cust_t See trd_rpt_cust_t sub structure.

Variable	Description
ask_side	trd_rpt_cust_t See trd_rpt_cust_t sub structure.

13.3.23.1 trd_rpt_cust_t

Variable	Description
party	party_t This is the declared counter party for this part of the trade. See party_t sub structure.
ex_client_s	char[10] Client – a free text field typically used to indicate the participant the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant 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.
open_close_req_c	uint8_t Ignore. Currently not used.
exch_order_type_n	uint16_t Exchange specific order types. Ignore any values returned that are not 2 = Short Sell order.
give_up_member	give_up_member_t See give_up_member_t sub structure.
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.
short_sell_quantity_i	int64_t Quantity of the trade that is short (partial or whole). 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.

13.3.24 exchange_info_t (named structure 50004)

Variable	Description
exchange_info_s	char[32] A free text field used at the participant's discretion.

13.3.25 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.

Variable	Description
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.26 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 sub structure.
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 Ignore. Currently not used.
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.27 single_order_insert_t (named structure 34808)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t The series for which the order exists.
items_n	uint16_t The number of sub-items following this top structure.
size_n	uint16_t The total size of the message, including this header.

13.3.28 single_order_update_t (named structure 34809)

Variable	Description
transaction_type	transaction_type_t Contains the transaction identifier that caused the BO5 to be sent.
series	series_t The series for which the order exists.
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

Variable	Description
	The number of sub-items following this top structure.
size_n	uint16_t The total size of the message, including this header.

13.3.29 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.
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.
exch_order_type_n	uint16_t Exchange specific order types. Ignore any values returned that are not in the list below: 2 = Short Sell order <i>(premium_i = an integer, order_type_c = 1 = Limit order premium_i = 0, order_type_c = 2 = Market order premium_i = 0, order_type_c = 3 = Market-to-Limit order premium_i = 0, order_type_c = 17, time_validity_n != 0 = Best-Limit order premium_i = an integer, order_type_c = 65 = Imbalance Limit order).</i>

Variable	Description
	<p>Can also be combined with other exchange specific order types/attributes) outlined below.</p> <p>4 = Market Bid order (<i>premium_i = an integer, order_type_c = 1, only entered by Market Control</i>).</p> <p>8 = Price Stabilisation/Green Shoe Order (<i>premium_i = an integer, order_type_c = 1, time_validity_n != 0</i>).</p> <p>32 = Undisclosed (use order_type_c to determine order type)</p> <p>64 = Centre Point Order (use order_type_c to determine Market or Limit)</p> <p>2048 = Sweep order (user order_type_c to determine Limit or Market-to-limit)</p> <p>4096 = Centre Point Block order (use order_type_c to determine Market or Limit).</p>
order_type_c	<p>uint8_t</p> <p>Order type – a combination of this field and the premium_i field signifies different types of orders. Possible values:</p> <p>1 = Limit price order (premium_i = an integer)</p> <p>2 = Market order (premium_i = 0)</p> <p>3 = Market-to-Limit order (premium_i = 0)</p> <p>17 = Best-Limit order (premium_i = 0 and time_validity_n != 0)</p> <p>65 = Imbalance Limit order (premium_i = an integer).</p> <p>Used in conjunction with values in the field <code>exch_order_type_n</code> to determine Undisclosed, Sweep, Centre Point and Centre Point Block orders.</p>
bid_or_ask_c	<p>uint8_t</p> <p>Bid or Ask. Possible values:</p> <p>1 = Buy</p> <p>2 = Sell.</p>
filler_2_s	<p>char[2]</p> <p>Ignore. Used for byte alignment.</p>

13.3.30 reserve_order_t (named structure 34812)

Variable	Description
display_quantity_i	<p>int64_t</p> <p>The shown quantity of an iceberg order. The <code>reserve_order_t</code> sub-structure only needs to be specified when entering iceberg orders.</p> <p>Can also be used for Limit Sweep order types to enter them as an iceberg order.</p>
original_display_quantity_i	<p>int64_t</p> <p>The original shown quantity of the iceberg orders, when it was first entered.</p>

13.3.31 basic_order_update_t (named structure 34815)

Variable	Description
premium_i	<p>int32_t</p> <p>The price of the order. A combination of this field and the <code>order_type_c</code> field signify different types of orders.</p> <p>0 = Market order.</p> <p>Any other value than zero = Limit order (<code>order_type_c = 1 or 65</code>).</p> <p>The price for a TMC order can be positive, zero or negative.</p>
quantity_i	<p>int64_t</p> <p>Quantity of the order.</p>

Variable	Description
	<p>When amending an order, this field can be set to:</p> <ul style="list-style-type: none"> the new value (delta_quantity_c = 1) the amount by which it should be changed (delta_quantity_c = 2) zero to indicate 'no change'. <p>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.</p>
balance_quantity_i	<p>int64_t</p> <p>Indicates that the participant wants a check done on the quantity of the existing order before amendment. Possible values:</p> <p>0 = No balance check is performed</p> <p>Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.</p>
time_validity_n	<p>uint16_t</p> <p>This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte).</p> <p>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.</p> <p>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
exch_order_type_n	<p>uint16_t</p> <p>Exchange specific order types. Ignore any values returned that are not in the list below:</p> <p>2 = Short Sell order</p> <p><i>(premium_i = an integer, order_type_c = 1 = Limit order</i></p> <p><i>premium_i = 0, order_type_c = 2 = Market order</i></p> <p><i>premium_i = 0, order_type_c = 3 = Market-to-Limit order</i></p> <p><i>premium_i = 0, order_type_c = 17, time_validity_n != 0 = Best-Limit order</i></p> <p><i>premium_i = an integer, order_type_c = 65 = Imbalance Limit order)</i></p> <p>Can also be combined with other exchange specific order types/attributes) outlined below.</p> <p>4 = Market Bid order</p> <p><i>(premium_i = an integer, order_type_c = 1, only entered by ASX Trading Operations).</i></p> <p>8 = Price Stabilisation/Green Shoe Order</p> <p><i>(premium_i = an integer, order_type_c = 1, time_validity_n != 0).</i></p> <p>32 = Undisclosed (use order_type_c to determine order type)</p> <p>64 = Centre Point Order (use order_type_c to determine Market or Limit)</p> <p>2048 = Sweep order (user order_type_c to determine Limit or Market-to-limit)</p>

Variable	Description
	4096 = Centre Point Block order (use order_type_c to determine Market or Limit).
delta_quantity_c	uint8_t Usually, on an order amendment, this field indicates if the quantity values are the new values or the changed values supplied for each order. However, for any transactions that can submit either an absolute or a delta quantity, the BO5 will return only the resulting absolute quantity. Possible value: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.

13.3.32 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. 0 = not defined 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 6 = Any Price Block order with mid-tick attribute set to on. Limit Sweep orders fully integrate the liquidity in Centre Point and ASX TradeMatch and will interact with '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 0, 1 or 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. 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.

Variable	Description
	This variable cannot be amended.
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 = minimum acceptable quantity must be satisfied in a single fill 2 = minimum acceptable quantity may be satisfied in multiple fills (aggregated execution).
filler_1_s	char[1] Ignore. Used for byte alignment.

13.3.33 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.

13.3.34 order_submitter_t (named structure 34819)

Variable	Description
submitter	trading_code_t See trading_code_t sub structure.

13.3.35 order_owner_t (named structure 34804)

Variable	Description
owner	trading_code_t See trading_code_t sub structure.

13.3.36 ranking_time_t (named structure 34949)

Variable	Description
timestamp_ranking	time_spec_t Ranking timestamp for Centre Point orders. See time_spec_t sub structure.

13.3.37 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.

Variable	Description
	Setting this field to zero for an order means ‘no Unintentional Crossing Prevention’ for this order.

13.3.38 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.

13.3.39 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 (exch_order_type_n != 2), must be set to 0, or this sub-structure should not be included at all. For orders that are short sell orders (exch_order_type_n = 2), must be equal to or less than the total order quantity and greater than zero.

13.3.40 short_sell_order_change_t (named structure 34830)

Variable	Description
short_sell_quantity_i	int64_t Partial short quantity of a short sell order.

13.3.41 enhanced_cp_matching_t (named structure 34831)

Variable	Description
participant_order_attribute_i	uint32_t Reserved for future use.
counter_order_attributes_i	uint32_t Reserved for future use.

13.3.42 ex_user_code_t

The structure identifies the user who placed this order on behalf of another user.

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 ASX 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]

Variable	Description
	The unique identifier of the user who placed the order. Participants can retrieve their own identifier using the <code>omniapi_get_info_ex()</code> function.

13.3.43 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 participant can be retrieved from the <code>clearing_customer_s</code> 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.

13.3.44 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.
premium_i	int32_t The price of the order. A combination of this field and the <code>order_type_c</code> field signify different types of orders. 0 = Market order (<code>order_type_c > 1</code>) Any value = Fixed price order (<code>order_type_c=1</code>). Note that the price for a TMC order can be positive, zero or negative.
block_n	uint32_t Block size. Possible values: 0 = Fill Or Kill order (<code>time_validity_n = 0</code>) 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 <code>block_n = 0</code> . Fill And Kill when <code>block_n = 1</code> . 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. Note: 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.

Variable	Description
	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. Ignore any values returned that are not in the list below: 2 = Short Sell order (<i>premium_i = an integer, order_type_c = 1 = Limit order</i> <i>premium_i = 0, order_type_c = 2 = Market order</i> <i>premium_i = 0, order_type_c = 3 Market-to-Limit order</i> <i>premium_i = 0, order_type_c = 17, time_validity_n != 0 = Best-Limit order</i>). Can also be combined with other exchange specific order types / attributes) outlined below. Can only be a sell order (bid_or_ask_c = 2). 4 = Market Bid order (<i>premium_i = an integer, order_type_c = 1, only entered by ASX Trading Operations</i>). 8 = Price Stabilisation/Green Shoe Order (<i>premium_i = an integer, order_type_c = 1, time_validity_n != 0</i>). 32 = Undisclosed (use order_type_c to determine order type) 64 = Centre Point Order (use order_type_c to determine Market or Limit) 2048 = Sweep order (user order_type_c to determine Limit or Market-to-limit) 4096 = Centre Point Block order (use order_type_c to determine Market or Limit).
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible value: 0 = Ignore, not relevant. Refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> for more information.
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) 2 = Market order (premium_i = 0) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0, time_validity_n != 0) 65 = Imbalance Limit order (premium_i = an integer).

Variable	Description
	Used in conjunction with values in the field <code>exch_order_type_n</code> to determine Undisclosed, Centre Point, Centre Point Block and Sweep orders.
<code>stop_condition_c</code>	<code>uint8_t</code> Ignore. Currently not used.
<code>filler_2_s</code>	<code>char[2]</code> Ignore. Used for byte alignment.

13.3.45 party_t

Variable	Description
<code>country_id_s</code>	<code>char[2]</code> For ASX Trade this is always set to "AU", indicating the Australian exchange.
<code>ex_customer_s</code>	<code>char[5]</code> 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 <code>country_id_s</code> field and this field uniquely define a trading participant.
<code>filler_1_s</code>	<code>char[1]</code> Ignore. Used for byte alignment.

13.3.46 trading_code_t

Variable	Description
<code>country_id_s</code>	<code>char[2]</code> ASX Trade this is always set to "AU", indicating the Australian exchange.
<code>ex_customer_s</code>	<code>char[5]</code> 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 <code>country_id_s</code> field and this field uniquely define a trading participant.
<code>user_id_s</code>	<code>char[5]</code> The unique identifier of an ASX Trade user. Users can retrieve their own identifier using the <code>omniapi_get_info_ex()</code> function.

13.3.47 originator_trading_code_t

Variable	Description
<code>country_id_s</code>	<code>char[2]</code> For ASX Trade this is always set to "AU", indicating the Australian exchange.
<code>ex_customer_s</code>	<code>char[5]</code> 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 <code>country_id_s</code> field and this field uniquely define a trading participant.
<code>user_id_s</code>	<code>char[5]</code>

Variable	Description
	The unique identifier of an ASX Trade user. Users can retrieve their own identifier using the <code>omniapi_get_info_ex()</code> function.

13.3.48 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since the Epoch (1970-01-01 00:00:00 UTC).
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

13.3.49 asx_exchange_info_t

Variable	Description
trade_report_info_s	char[16] Free text field.
boq_list_s	char[6] List of up to 3 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: 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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

13.3.49.1 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.

Variable	Description	Character Position	ASIC defined content
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 BO10 Price Information

14.1 Broadcast Function

This broadcast provides information on changes in the equilibrium price (Indicative Auction Price - IAP). Each broadcast contains a list of updated series where all series belong to the same instrument class. The broadcasts are subjected to a hold back period, currently set to 500 milliseconds.

In ASX Trade, when an equilibrium price does not exist for a particular series, the following fields in the BO10 transaction contain:

- `equilibrium_price_i` – the 31st bit (highest bit) is set and all other bits are set to zero
- `equilibrium_quantity_i` – zero
- `best_bid_premium_i` – the current best bid price
- `best_ask_premium_i` – the current best ask price
- `best_bid_quantity_i` – quantity available at the best bid
- `best_ask_quantity_i` – quantity available at the best ask.

14.2 Broadcast Properties

Function Call	<code>omniapi_read_event_ext_ex</code> or <code>omniapi_read_event_block</code>
Struct Name	<code>equil_price_update_t</code>
Information Type	Instrument Class

14.3 Message Structure

14.3.1 `equil_price_update_t`

Variable	Description
<code>broadcast_type</code>	<code>broadcast_type_t</code> Contains the following: {'B', 'O', 10}.
<code>items_n</code>	<code>uint16_t</code> Identifies how many items are held in the item array.
<code>filler_2_s</code>	<code>char[2]</code> Ignore. Used for byte alignment.
<code>item</code>	<code>equil_price_update_item_t</code> Array of items – maximum 35 items. Refer to <code>equil_price_update_item_t</code> sub structure.

14.3.2 `equil_price_update_item_t`

Variable	Description
<code>series</code>	<code>series_t</code> Used to identify the series for this item.
<code>equilibrium_quantity_i</code>	<code>int64_t</code> The quantity that is executed at the equilibrium price once the instrument series goes into a trading state where order matching is enabled.
<code>equilibrium_price_i</code>	<code>int32_t</code>

Variable	Description
	<p>The equilibrium price (IAP) of the instrument series, i.e. the price that the series would trade at when it goes into a trading state where order matching is enabled.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p> <p>Note: If matching_price_type_c is 2, then this value indicates a reference price.</p>
best_bid_premium_i	<p>int32_t</p> <p>The best bid price that will be in the order book when the instrument series goes into a trading state where order matching is enabled.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
best_ask_premium_i	<p>int32_t</p> <p>The best ask price that will be in the order book when the instrument series goes into a trading state where order matching is enabled.</p> <p>Decimal places are implied from the attributes of the associated instrument class to the series.</p>
best_bid_quantity_i	<p>int64_t</p> <p>The quantity for the best bid price that will be in the order book when the instrument series goes into a trading state where order matching is enabled. If the best_ask_premium_i and the equilibrium_price_i values are equal then this value will be the surplus volume.</p>
best_ask_quantity_i	<p>int64_t</p> <p>The quantity for the best ask price that will be in the order book when the instrument series goes into a trading state where order matching is enabled. If the best_ask_premium_i and the equilibrium_price_i values are equal then this value will be the surplus volume.</p>
matching_price_type_c	<p>uint8_t</p> <p>Different type of prices distributed as equilibrium price.</p> <p>Possible values:</p> <p>1 = Normal indicative Equilibrium Price</p> <p>2 = Reference price. Not currently used</p>
filler_3_s	<p>char[3]</p> <p>Ignore. Used for byte alignment.</p>

15 BO14/BO15 Order Book with Levels

15.1 Broadcast Function

These broadcasts provide summary information on changes in the order book. Only the total volume for each price is given and no order related information is included. The broadcasts are subjected to a hold back period, currently set to 500 milliseconds.

BO14 and BO15 are interchangeable broadcasts, but with separate configurations. BO14 provides the top five levels and BO15 provides only the top level order book information. BO14 is utilised for Derivatives markets (Stock Derivatives market and Index Derivatives market) and the Listed Funds, Structured Products & Warrants market. BO15 is utilised for Equity and Interest Rate markets. The two different subscription alternatives are provided catering to differing bandwidth utilisation.

In order to maintain a real time picture of the BO14 information, the OI application must use IQ18 to download a baseline of the information. In order to maintain a real time picture of the BO15 information the Open Interface application must use IQ19 to download a baseline of the information. The sequence for this is described in the IQ18/IQ19 queries of this document (refer to *IQ18 Total Volumes and Prices in ASX Trade Queries*).

The broadcast is a Variable Information Broadcast (VIB) containing one broadcast header structure followed by one or more structures identified by their Named Structure Number property.

15.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead it holds a sequence of possible structs.
Information Type	Instrument Class

15.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_hdr_t
- one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - ob_levels_order_number_t (named structure 33004)
 - ob_levels_sequence_number_t (named structure 33001)
 - ob_levels_total_quantity_t (named structure 33005)
 - ob_levels_no_of_orders_t (named structure 33033)
 - ob_levels_price_volumes_t (named structure 33003)
 - ob_levels_id_t (named structure 33002)
 - ob_levels_undisclosed_quantity_t (named structure 33041).

15.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t

Variable	Description
	Contains the following: {'B', 'O', 14} or {'B', 'O', 15}.
items_n	uint16_t The number of sub-items following this header.
size_n	uint16_t The total size of the message including this header.

15.3.2 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 The size of the structure that is to follow, including this header.

15.3.3 ob_levels_sequence_number_t (named structure 33001)

This structure is always present as the first variable structure in a BO14/BO15 broadcast. It occurs exactly once in the broadcast. It should not be processed by the application.

Variable	Description
sequence_number_u	uint32_t Ignore. Currently not used.

15.3.4 ob_levels_order_number_t (named structure 33004)

This structure is only present if the values of either fields have changed, i.e. if either the best ranked order number of the bid or ask (or both) has changed.

Variable	Description
order_number_bid_u	quad_word The order number for the first ranked bid order in the order book.
order_number_ask_u	quad_word The order number for the first ranked ask order in the order book.

15.3.5 ob_levels_total_quantity_t (named structure 33005)

Variable	Description
total_quantity_bid_u	int64_t Total bid quantity (excluding hidden quantities).
total_quantity_ask_u	int64_t Total ask quantity (excluding hidden quantities).

15.3.6 ob_levels_price_volumes_t (named structure 33003)

The price masks are interpreted as bit fields where currently the low 10 bits are used. Bit 0 corresponds to the first ranked price, bit 1 to the second best ranked price, etc. For each bit set in the mask an array item is present. Bid items are placed before ask items in the array. Better ranked prices are placed before lower ranked prices in the array. The *items_c* field holds the total number of items within the array.

For ASX Trade, the current configuration for this response has been made so that only five levels of depth are disseminated. However users should be aware that the system is capable of 10 levels of depth in this response. The number of levels are indicated in the `premium_levels_c` field.

Also, ASX Trade only allows either `ob_levels_price_t` (named structure 33006) or `ob_levels_price_volumes_t` (named structure 33003) to be distributed for a given instrument. They are never distributed simultaneously. Currently, only the `ob_levels_price_volumes_t` struct is distributed.

Variable	Description
<code>bid_mask_n</code>	<p><code>uint16_t</code> Interpreted as a bit field where the low 10 bits are used. Bit 0 corresponds to the first ranked price, bit 1 to the second best ranked price, etc. For each bit set in the mask, an associated array item is present in this structure. The bit set indicates to which price level the array item corresponds. All bid items are placed before any ask items in the array. Better ranked prices are placed before lower ranked prices in the array. The items field holds the total number of items within the array.</p> <p>Example: If the bid price mask has a value of 3 (bit 0 and 1 set) and the ask price mask has a value of 8 (bit 3 set only), the broadcast will hold 3 array items consisting of:</p> <ul style="list-style-type: none"> • Array[0]: Price and quantity for bid level 1 • Array[1]: Price and quantity for bid level 2 • Array[2]: Price and quantity for ask level 4. <p>This broadcast affectively indicates that the bid levels of 3, 4, and 5, and the ask levels of 1, 2, 3 and 5 have not changed since the previous broadcast.</p> <p>Example: If the bid price mask has the value 0 and the ask price mask has the value 31 (bit 0 to 4 set), the array consists of the following items:</p> <ul style="list-style-type: none"> • Array[0]: Price and quantity for ask level 1 • Array[1]: Price and quantity for ask level 2 • Array[2]: Price and quantity for ask level 3 • Array[3]: Price and quantity for ask level 4 • Array[4]: Price and quantity for ask level 5. <p>This broadcast indicates that there has been no change on the bid side for this series since the previous broadcast.</p> <p>Example: If the bid price mask has the value 16 (bit 4 set) and the ask price mask has the value 24 (bit 3 and 4 set), the array consists of the following items:</p> <ul style="list-style-type: none"> • Array[0]: Price and quantity for bid level 5 • Array[1]: Price and quantity for ask level 4 • Array[2]: Price and quantity for ask level 5. <p>This broadcast also informs the user that there has been no change to levels 1, 2, 3 and 4 on the bid side, and no change to levels 1, 2, and 3 on the ask side since the previous broadcast.</p>
<code>ask_mask_n</code>	<p><code>uint16_t</code> Refer to the description for <code>bid_mask_n</code>.</p>
<code>premium_levels_c</code>	<p><code>uint8_t</code> Propagates the currently distributed price depth for this instrument series. Possible values are currently in the range from 0 to 5. This value can be changed up to a maximum of 10. 0 = the exchange doesn't distribute any prices at all 1 = the exchange distributes the first ranked price level</p>

Variable	Description
	2 = the exchange distributes two best prices levels 3 = the exchange distributes three best prices levels 4 = the exchange distributes four best prices levels 5 = the exchange distributes five best prices levels. The premium levels could be changed during the day for a given instrument series. In the case where the premium level is decreased, the application must clear all price levels beyond the current level.
demands_populated_c	uint8_t Indicates if the distribution of quantity is enabled or disabled for the different price levels. For this struct the value is always one (enabled) as the volume will be included in the items.
items_c	uint8_t Indicates how many items are in the array below.
filler_1_s	char[1] Ignore. Used for byte alignment.
item	ob_levels_price_volumes_item_t[32] See ob_levels_price_volumes_item_t sub structure.

15.3.7 ob_levels_price_volumes_item_t

Variable	Description
premium_i	int32_t The price for this level. If the 31 st bit (highest bit) is set and the rest are zero, then this indicates that there is no price available. This differs from the value of zero (all bits zero) indicating a price of zero. ASX Trade allows orders to be placed with a price of zero for combinations. Decimal places are implied from the attributes of the associated instrument class to the series.
demand_u	int64_t Total quantity of orders (excluding hidden quantities).

15.3.8 ob_levels_no_of_orders_t (named structure 33033)

Users are advised that it is certainly possible to receive different values in the bid_mask_n and ask_mask_n fields compared to those in the ob_levels_price_volume_t structure.

Variable	Description
bid_mask_n	uint16_t Interpreted as a bit field where the low 10 bits are used. Bit 0 corresponds to the first ranked price, bit 1 to the second best ranked price, etc. For each bit set in the mask an associated array item is present in this structure. The bit set indicates to which price level the array item corresponds. All bid items are placed before any ask items in the array. Better ranked prices are placed before lower ranked prices in the array. For an example of how to use this field refer to ob_levels_price_volumes_t.
ask_mask_n	uint16_t Refer to the description for bid_mask_n above.

Variable	Description
total_no_of_bid_orders_u	uint32_t Total number of all the bid orders for the given series.
total_no_of_ask_orders_u	uint32_t Total number of all the ask orders for the given series.
premium_levels_c	uint8_t Propagates the currently distributed order book depth for this instrument series. Possible values are currently in the range from 0 to 5. This value can be changed up to a maximum of 10. 0 = the exchange doesn't distribute any prices at all 1 = the exchange distributes the first ranked price level 2 = the exchange distributes two best prices levels 3 = the exchange distributes three best prices levels 4 = the exchange distributes four best prices levels 5 = the exchange distributes five best prices levels. The premium levels could be changed during the day for a given instrument series. In the case where the premium level is decreased, the application must clear all price levels beyond the current level.
filler_2_c	char[2] Ignore. Used for byte alignment.
items_c	uint8_t Indicates how many items are in the array below.
item	ob_levels_no_of_orders_item_t[32] See ob_levels_no_of_orders_item_t sub structure.

15.3.9 ob_levels_no_of_orders_item_t

Variable	Description
no_of_orders_u	uint32_t The number of orders for this price level.

15.3.10 ob_levels_id_t (named structure 33002)

This structure defines the instrument series to which succeeding variable structures relate (up until the occurrence of a new ob_levels_id_t structure).

Example:

```

...      (Previous series)
ob_levels_id      Sets series A
ob_levels_price_volumes  Prices and volumes for series A
ob_levels_order_number  Order numbers for series A OB Levels
ob_levels_id      Sets series B
ob_levels_order_number  Order numbers for series B OB Levels
ob_levels_id      Sets series C

```

ob_levels_price_volumes Prices and volumes for series C

... (Succeeding series)

Variable	Description
series	series_t Contains the series to which this price information relates.
block_n	uint32_t Ignore. Always set to one.

15.3.11 ob_levels_undisclosed_quantity_t (named structure 33041)

This structure will show whether a price level, either side of the book, has Undisclosed Quantity or not. However, the actual undisclosed quantities are not revealed. It is possible for a price level to have the disseminated quantity zero.

Variable	Description
bid_mask_n	uint16_t Used as a bit mask in the same way as described in the bid_mask_n field of the ob_levels_price_volumes_t (named structure 33003).
ask_mask_n	uint16_t Used as a bit mask in the same way as described in the bid_mask_n field of the ob_levels_price_volumes_t (named structure 33003).

16 BO38 Market Maker Protection Settings Information

16.1 Broadcast Function

This broadcast disseminates the calculated Quantity and Delta Protection values for a particular underlying when Market Maker Protection is triggered.

16.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	market_maker_protection_info_t
Information Type	Dedicated

16.3 Message Structure

16.3.1 market_maker_protection_info_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'O', 38}.
trading_code	trading_code_t Indicates the Market Maker's participant identifier. See trading_code_t sub structure.
series	series_t The series to which this information relates.
calc_quantity_protection_q	int64_t Calculated quantity value for Market Maker Protection.
calc_delta_protection_q	int64_t Calculated delta value for Market Maker Protection.

16.3.2 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 vendors have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely defines a trading participant.
user_id_s	char[5] The unique identifier of the user who placed the order. Not used in this broadcast.

17 BO99 Block Transaction Response

17.1 Broadcast Function

This broadcast is sent when a block quote (MO36) is only partly successful. The response holds detailed information on the part or parts of the block transaction that failed.

This information is also disseminated in a BO5 broadcast. Users who are permissioned to receive BO5s will receive the information in that broadcast. Users who do not get the BO5 broadcasts (e.g. Market Makers only receive BO5 broadcasts for quotes when they trade or are deleted) will receive this BO99 broadcast.



Note:

The BO99 broadcast is not sent if the whole block quote transaction is rejected.

If either side of an order book in a block quote fails, the affected quote fails and the relevant item in the array has transaction_status_i aborted.

17.2 Broadcast Propertie

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	block_order_response_t
Information Type	Dedicated

17.3 Message Structure

17.3.1 block_order_response_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'O', 99}.
order_number_u	quad_word Indicates the order identifier.
items_c	uint8_t Number of items in the array.
filler_3_s	char[3] Ignore. Used for byte alignment.
item	block_order_response_item_t[100] See block_order_response_item_t sub structure below.

17.3.2 block_order_response_item_t

Variable	Description
transaction_status_i	int32_t Contains the same information as returned from the omniapi_tx_ex function, indicating the status. Possible values: 0 = Successful Any other value = Transaction aborted.

Variable	Description
trans_ack_i	int32_t Contains the same information as the transaction status from the omnapi_tx_ex function. The codes vary depending on the context in which they occur, but some common examples would be: 1 = No part of the order placed in the order book and no part closed 2 = The whole order closed 3 = The order partially closed and nothing placed in the order book 4 = The whole order placed in the order book 6 = The order partially placed in the order book and partially closed. GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type. MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.
item_number_c	uint8_t The item number in a quote transaction with a value of zero representing the first item. Note: This does not indicate which side of the quote failed. It could be either the bid or the ask or both.
filler_3_s	char[3] Ignore. Used for byte alignment.

18 BU28 Central Group Update

18.1 Broadcast Function

This broadcast is sent when a new central group has been defined in ASX Trade or an existing central group has been modified.

18.2 Broadcast Properties

Function call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct name	central_group_update_t
Information type	General

18.3 Message Structure

18.3.1 central_group_update_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 28}.
chg_type_n	uint16_t The type of update performed. Possible values: 1 = Add 2 = Delete 3 = Alter.
filler_2_s	char[2] Ignore. Used for byte alignment.
da28	da28_t See da28_t sub structure below.

18.3.2 da28_t

Variable	Description
central_group_s	char[12] The ASCII representation of a centrally defined group.
segment_number_n	uint16_t Used if the whole central group cannot be placed in one broadcast. If not, all series can be sent, the segment number is incremented with one until the whole Central Group is distributed. The last broadcast is sent with segment number = 0.
items_n	uint16_t The number of items held in the array.
item	item[30] See item sub structure for da28_t below.

18.3.2.1 item struct for da28_t

Variable	Description
long_ins_id_s	char[32] Additional instrument series information in ASCII characters.
leg_number_n	uint16_t The leg number of the central group leg.
sort_type_c	uint8_t Not applicable, ignore.
filler_1_s	char[1] Ignore. Used for byte alignment.

19 BU53 Corporate Action Update

19.1 Broadcast Function

This broadcast is sent when a new corporate action code is added during the day. One broadcast can contain several items.

19.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex
Struct Name	corp_action_update_bu53_t
Information Type	General

19.3 Message Structure

19.3.1 corp_action_update_bu53_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 53}.
chg_type_n	uint16_t Ignore. Not used.
items_n	uint16_t The number of items held in the array.
item	corp_action_update_bu53_item_t[50] See corp_action_update_bu53_item_t sub structure below.

19.3.2 corp_action_update_bu53_item_t

Variable	Description
da53	da53_t See da53_t sub structure below.

19.3.3 da53_t

Variable	Description
series	series_t This is filled in with different information depending to which level the corporate action code is assigned (see corp_action_level_c variable): <ul style="list-style-type: none"> Linked Underlying level - Only a value in commodity_n is filled in, the rest of the fields are zero. All underlyings connected to this linked underlying are affected. The linked underlying is distributed in the answer of DQ120, Query Underlying. Refer to <i>ASX Trade Queries</i>. Underlying level - Only a value in commodity_n is filled in, the rest of the fields are zero.

Variable	Description
	<ul style="list-style-type: none"> Instrument Class level - A value in country_c, market_c, instrument_group_c and commodity_n. Instrument Series level - A complete series definition.
corp_action_code_s	char[2] Corporate Action Code.
corp_action_type_c	uint8_t Indicates if a corporate action code is a basis of quotation or a status note. Possible values: 1 = Basis of Quotation 2 = Status Note.
corp_action_status_c	uint8_t It is possible to remove (disable) a corporate action code during the day. Possible values: 1 = Enabled 2 = Disabled.
corp_action_level_c	uint8_t The instrument level of the corporate action: Possible values: 1 = Underlying 2 = Linked Underlying 3 = Instrument Class 4 = Instrument Series.
filler_3_s	char[3] Ignore. Used for byte alignment.

20 BU56 Special Liability Update

20.1 Broadcast Function

This broadcast is sent to a dedicated user when the Special Liability table connected to this user is changed. When a Special Liability table that is connected to more than one user is updated in ASX Trade, all connected users will receive a dedicated broadcast (i.e. if 10 users are connected, 10 dedicated broadcasts are sent).

Special Liability limits can be set for each market by ASX Trading Operations on the advice from the participant compliance officer.

Participants are not permitted to enter orders that exceed the value specified by the Special Liability. Note that these values need to be enforced by the OI application. The back end does not check these limits prior to an order being submitted.

20.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	special_liability_update_bu56_t
Information Type	Dedicated

20.3 Message Structure

20.3.1 special_liability_update_bu56_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 56}.
chg_type_n	uint16_t The type of change indicated in this broadcast. Possible values: 1 = Add 2 = Delete 3 = Amend.
items_n	uint16_t The number of items held in the array.
trading_code	trading_code_t See trading_code_t sub structure below.
da56	da56_t See da56_t sub structure below.

20.3.2 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]

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.
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.

20.3.3 da56_t

Variable	Description
item	internal item structure [150] See internal item sub structure below.

20.3.4 item struct for da56_t

Variable	Description
spec_liab_limit_i	int32_t Special liability limit for each market.
country_c	uint8_t Always 15.
market_c	uint8_t An integer representing the market code. For an entire list of possible values refer to ASX Trade Markets, Instrument Groups and Trade Condition Codes .
filler_2_s	char[2] Ignore. Used for byte alignment.

21 BU87 Market Maker Protection Update

21.1 Broadcast Function

This broadcast is sent when the Market Maker Protection parameters have been updated for a particular underlying.

21.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	mm_protection_update_t
Information Type	Dedicated

21.3 Message Structure

21.3.1 mm_protection_update_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 87}.
chg_type_n	uint16_t The type of change indicated in this broadcast. Possible values: 1 = Add 2 = Delete 3 = Amend.
filler_2_s	char[2] Ignore. Used for byte alignment.
da87	da87_t See da87_t sub structure below.

21.3.2 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 0 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 0 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.

Variable	Description
	<p>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.</p> <p>A value of 0 means that Market Maker Protection functionality is turned off for the applicable underlying.</p>
frozen_time_i	<p>int32_t</p> <p>Specifies the time interval in milliseconds when quotes are rejected after Market Maker Protection has been triggered.</p> <p>A value of 0 means that quotes are considered as frozen for the rest of the day.</p>
commodity_n	<p>uint16_t</p> <p>The code for the underlying that the Market Maker Protection parameters returned in this item apply to.</p>
country_id_s	<p>char[2]</p> <p>For ASX Trade this field is always set to "AU", indicating the Australian exchange.</p>
ex_customer_s	<p>char[5]</p> <p>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.</p>
include_futures_c	<p>uint8_t</p> <p>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).</p> <p>Possible values:</p> <p>1 = Yes, include Futures and Forwards in the Delta Protection calculation</p> <p>2 = No, do not include Futures and Forwards in the Delta Protection calculation.</p>
filler_2_s	<p>char[2]</p> <p>Ignore. Used for byte alignment.</p>

22 BU120 Delta Underlying Update

22.1 Broadcast Function

This broadcast is used to send out information about a new underlying or an underlying that has changed.



Note:

ASX Trading Operations current practice is to add, delete and amend underlyings overnight and not during the trading day. However the definition of this broadcast is provided here in the rare case that an underlying is changed intra-day.

22.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	This is a Variable Information Broadcast and has no topmost structure. Instead, the sequence of possible structs are described in the Message Structure section below.
Information Type	General

22.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_segment_hdr_t
- one of:
 - item_hdr_t
 - one of:
 - sub_item_hdr_t
 - one of:
 - ns_delta_header_t (named structure 37001)
- one or more sequences of:
 - item_hdr_t
 - one or more sequences of:
 - sub_item_hdr_t
 - a choice either of:
 - ns_remove_t (named structure 37002)
 - or a set of the following:
 - ns_underlying_basic_t (named structure 37201)
 - ns_fixed_income_t (named structure 37202)
 - ns_coupon_dates_t (named structure 37203).

22.3.1 broadcast_segment_hdr_t

Currently in ASX Trade, out of all the broadcasts that contain this header, only the BU120 uses the segment_number_n field. It indicates that information relating to one entity has been split over two or more broadcasts.

The maximum size of a broadcast is approximately 14 KB. Although unlikely, it is still theoretically possible for the underlying of an interest rate to contain up to 700 different coupons. This amount of information cannot be contained in the one broadcast and will therefore be divided over several broadcasts.

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 120}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The total size of the message, including this header.
segment_number_n	uint16_t A BU120 broadcast can be segmented if the one broadcast contains too much information to be sent over the OMdu at once. This field indicates the number of the segment, starting at one for the first segment, two for the second, and so on with a value of zero indicating the last segment. For single segmented broadcasts, the value in this field is zero.
filler_2_s	char[2] Ignore. Used for byte alignment.

22.3.2 item_hdr_t

Variable	Description
items_n	uint16_t The number of sub-items following this item header.
size_n	uint16_t The total size of the following sub-items, including this header.

22.3.3 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 The size of the structure that is to follow, including this header.

22.3.4 ns_delta_header_t – named struct number 37001

Variable	Description
download_ref_number_q	uint32_t The download reference number for this answer. The application needs to store this number so that it may use it again for the next time it makes a DQ120 query, and thereby only retrieve the delta information.
full_answer_timestamp	time_spec_t See time_spec_t sub structure below.
full_answer_c	uint8_t Indicates if a full answer has been returned. Possible values: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.

22.3.5 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since 1970-01-01 00:00:00 UTC.
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

22.3.6 ns_remove_t – named struct number 37002

Indicates that the underlying has been removed. Only this structure will be sent if the underlying has been removed.

Users are advised that due to the current Trading Operations practices, a BU120 will never contain *ns_remove_t*. All underlyings are removed overnight. Hence the *ns_remove_t* structure will only be seen with the response to the associated query – DQ120. However, this information has been left here for future reference.

Variable	Description
series	series_t The underlying that has been removed from the market.

22.3.7 ns_underlying_basic_t (named struct number 37201)

Variable	Description
commodity_n	uint16_t The code for this underlying. This is the value that is used in the series_t struct. Users need to be aware that after delisting an underlying the code can be reused again after seven years.
linked_commodity_n	uint16_t If an underlying is linked to another, the commodity code of the real underlying is placed in this field. A value of zero means that this underlying is not linked to any other.
state_number_n	uint16_t The binary representation of the trading state. Available values can be retrieved through the Trading State Query (refer to <i>DQ29 Trading State</i> in <i>ASX Trade Queries</i> for more information). This field may only be used when a new underlying is introduced. ASX Trading Operations have the ability to set its instrument session state at this point.
dec_in_price_n	uint16_t The decimals in the underlying price received from external sources.
com_id_s	char[6] The ASCII representation of the underlying.
isin_code_s	char[12] A code which uniquely identifies a specific securities issue (International Securities Identification Number - ISIN). The ISIN shall consist of: <ul style="list-style-type: none"> • A prefix, which is the alpha-2 country code • The basic number, which is nine characters • A check digit. For more information about ISIN code, see the international standard ISO 3166.
name_s	char[32] The ASCII description of the underlying.

Variable	Description
base_cur_s	char[3] Type of currency if price_unit_c = 1 (i.e. Price).
deliverable_c	uint8_t Defines if a series can be delivered or not (cash settlement): 1 = Yes 2 = No. A special case exists for warrants on this field. Users will be required to obtain the deliverable information from the Warrant Product Disclosure Statement and/or the ASX Signal E.
underlying_type_c	uint8t The type of underlying. Possible values: 1 = Stock 2 = Currency 3 = Interest Rate 4 = Energy 5 = Soft and Agricultures 6 = Metal 7 = Stock Index 8 = Currency Index 9 = Interest Rate Index 10 = Energy Index 11 = Soft and Agricultures Index 12 = Metal Index.
price_unit_c	uint8_t The price unit for the underlying. Possible values: 1 = Price 2 = Yield 3 = Points 4 = Yield Difference 5 = IMM Index 6 = Basis Points 7 = Inverted Yield 8 = Percentage of Nominal 9 = Dirty Price.
underlying_status_c	uint8_t 1 = Active 2 = Delisted.
underlying_issuer_s	char[6] The issuer of the underlying. This is typically used for warrants.
sector_code_s	char[4] Ignore. Currently not used.
virtual_c	uint8_t Ignore. Currently not used.
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ext_provider_c	char[1]

Variable	Description
	<p>External Price feed provider (S&P Index).</p> <p>N = NMF, S = Six, O = Six OMX, F = Direct Feed</p> <p>R = Direct Feed OPRA, T = Transaction, L = LMIL</p> <p>E = Reuter SSL.</p>
external_id_s	char[40] External Price feed identity.
cur_unit_c	uint8_t Specifies the currency unit for underlying prices. 1 = Primary Unit 2 = Secondary Unit 3 = Tertiary Unit. ASX Trade allows each currency up to three units of measurement. E.g., Australian dollars has a primary unit of “dollar”, a secondary unit of “cent”, and no tertiary unit.
db_operation_c	uint8_t Operation that ASX Trade has carried out on this particular item. Note: Participants are required to store in their OI applications a copy of the underlyings that concern them. An insert could appear for an item that they have already recorded or an update could appear for an item that they have yet to record. These situations need to be catered for by the participant. As mentioned previously, a remove will be indicated by the named structure <i>ns_remove_t</i> (named structure 37002). Possible values: 1 = Insert 2 = Update.
filler_3_s	char[3] Ignore. Used for byte alignment.

22.3.8 ns_fixed_income_t (named struct number 37202)

Variable	Description
nominal_value_q	int64_t Contains the nominal value (if any) of the underlying security.
coupon_interest_i	uint32_t Contains the coupon interest for the underlying. Only applicable for interest related underlyings.
dec_in_nominal_n	uint16_t Contains the implied decimals for the nominal_value_q field.
coupon_settlement_days_n	uint16_t Number of settlement days at coupon.
coupon_frequency_n	uint16_t Number of coupons per year.
rate_determ_days_n	uint16_t Specifies number of rate determination days.
date_release_s	char[8] Contains the issue date of the underlying if it has a limited lifetime. Normally only applicable for bonds. Format: YYYYMMDD.

Variable	Description
date_termination_s	char[8] Contains the maturity date of the underlying if it has a limited lifetime. Normally only applicable to bonds. Format: YYYYMMDD.
date_dated_s	char[8] Contains the date of the underlying from when the coupon interest is calculated. Format: YYYYMMDD.
date_proceed_s	char[8] Proceed date for fixed income underlying, YYYYMMDD. If the last payment falls on a non-business day, the payment and the maturity is pushed forward to the next business day, the so-called Proceeds Date.
fixed_income_type_c	uint8_t Type of fixed income security. Possible values: 0 = Not Applicable 1 = Bill 2 = Bond 3 = Index Linked Bonds 4 = Bond Floating 5 = Lottery Bond 6 = Convertible Bond 7 = Structured Bond 8 = Fixing 9 = Credit Certificates.
day_calc_rule_c	uint8_t Day Calculation Rule. Possible values: 1 = ACTACT 2 = ACTAFB 3 = EU30360 4 = US30360 5 = ACT365 6 = ACT360.
filler_2_s	char[2] Ignore. Used for byte alignment.

22.3.9 ns_coupon_dates_t (named struct number 37203)

Variable	Description
date_coupddiv_s	char[8] Coupon date for bond underlying or dividend date for stock underlying. Format: YYYYMMDD.
date_booksclose_s	char[8] Books close date for bond underlying. This is the date before the coupon date on which the owner must be registered to receive the coupon payment. Format: YYYYMMDD.
dividend_i	uint32_t The dividend for the stock.

23 BU122 Delta Instrument Class Update

23.1 Broadcast Function

This broadcast is used to send out information about a new instrument class or an instrument class that has changed.

Since this broadcast has an information type of “General” its subscription cannot be filtered nor can ASX tailor the configuration of its dissemination. Therefore subscribed users will receive broadcasts on classes to which they have no access. The OI application must handle this situation.



Note:

ASX Trading Operations current practice is to add, delete and amend instrument classes overnight and not during the trading day. However the definition of this broadcast is provided here in the rare case that a class is changed intra-day.

23.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	This is a Variable Information Broadcast and has no topmost structure. Instead, the sequence of possible structs are described in the Message Structure section below.
Information Type	General

23.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_segment_hdr_t
- one of:
 - item_hdr_t
 - one of:
 - sub_item_hdr_t
 - a choice either of:
 - ns_delta_hdr_t (named structure 37001)
- one or more sequences of:
 - item_hdr_t
 - one or more sequences of:
 - sub_item_hdr_t
 - a choice either of:
 - ns_remove_t (named structure 37002)
 - or a set of the following:
 - ns_inst_class_basic_t (named structure 37101)
 - ns_price_tick_t (named structure 37102)
 - ns_block_size_t (named structure 37103)
 - ns_calc_rule_t (named structure 37104)
 - ns_crossing_rule_t (named structure 37106)

- ns_inst_class_ext1_t (named structure 37107).

23.3.1 broadcast_segment_hdr_t

Currently in ASX Trade, out of all the broadcasts that contain this header, only the BU120 uses the *segment_number_n* field. It indicates that information relating to one entity has been split over two or more broadcasts. The header is provided in this broadcast in case it is needed in the future.

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 122}
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the structure that is to follow, including this header.
segment_number_n	uint16_t Ignore. Not used in this broadcast.
filler_2_s	char[2] Ignore. Used for byte alignment.

23.3.2 item_hdr_t

Variable	Description
items_n	uint16_t The number of sub-items following this item header.
size_n	uint16_t The total size of the following sub-items, including this header.

23.3.3 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 The size of the structure that is to follow, including this header.

23.3.4 ns_delta_hdr_t (named structure 37001)

Variable	Description
download_ref_number_q	int64_t Reference number to use in delta queries and answers. For more information, see <i>Working with an Open Interface - Programming Considerations, Delta Queries</i> in <i>ASX Trade Introduction and Business Information</i> . This number is always increasing, but may contain gaps. The application needs to store this number so that it may use it again for the next time it makes a DQ122 query, and thereby only retrieve the delta information.
full_answer_timestamp	time_spec_t See time_spec_t sub structure below.

Variable	Description
full_answer_c	uint8_t A full answer is enforced in the delta query. Possible values: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.

23.3.5 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since 1970-01-01 00:00:00 UTC.
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

23.3.6 ns_remove_t (named structure 37002)

Indicates that the class has been removed. Only this structure will be sent if the instrument class has been removed.



Note:

Participants are required to store in their OI applications a copy of the instrument classes that concern them. A remove structure could appear for an item that is unknown to them. Participants need to cater for this situation.

Variable	Description
series_t	series_t Contains the class that has been removed.

23.3.7 ns_inst_class_basic_t (named structure 37101)

Variable	Description
series_t	series_t Contains the instrument class to which this information relates.
upper_level_series_t	series_t Contains the instrument class from which the class defined in series_t derives. This field is only relevant when the instrument class in series_t is a derivate (i.e. where derivate_level_n > 0). E.g. for an option on a stock this field contains the instrument class of the stock.
price_quot_factor_i	int32_t Specifies the price quotation factor. E.g., a value of 100 indicates that the options of this instrument class are quoted in hundredths of an option. If the price of one option is 6.5 cents, and this field is 100, the actual payment for the option will be 6.5 * 100 = 650 cents.
contract_size_i	int32_t For an option this indicates the contract size, i.e., the number of units of the underlying security that concerns one unit of the instrument. E.g., buying 10 call options with contract size of 100 will give the purchaser the right to buy 10 * 100 = 1,000 of the underlying security.

Variable	Description
	<p>For a warrant, the contract size indicates the number of units of the warrant that concerns one unit of the underlying security. E.g., a value of four indicates that four warrants are equivalent to one underlying security.</p> <p>The exceptions to this rule are for index warrants where it represents the cash value per index point, and for basket warrants where it represents the minimum parcel of warrants required to be held for early exercise.</p>
redemption_value_i	<p>int32_t</p> <p>Redemption value represents the amount paid at maturity. The redemption value is equal to the nominal value except for securities with amortisation or options.</p> <p>The redemption value is expressed as a percentage of Nominal Value.</p> <p>The value is a decimal value stored with six decimals, e.g. 100% is stored as 1000000.</p>
undisclosed_min_ord_val_i	<p>int32_t</p> <p>Minimum order value for undisclosed quantity orders.</p> <p>The value is always expressed in the primary currency unit.</p> <p>The value is defined as quantity * price * price quotation factor.</p> <p>Assumes decimal places have not been applied.</p>
opt_min_ord_val_i	<p>int32_t</p> <p>Optional minimum order value.</p> <p>The value is always expressed in the primary currency unit. The value is defined as quantity * price * price quotation factor.</p> <p>Assumes decimal places have not been applied.</p>
opt_min_trade_val_i	<p>int32_t</p> <p>Optional minimum trade value.</p> <p>The value is always expressed in the primary currency unit.</p> <p>The value is defined as quantity * price * price quotation factor.</p> <p>Assumes decimal places have not been applied.</p>
derivate_level_n	<p>uint16_t</p> <p>The derivative level of the instrument. Possible values:</p> <p>0 = Stock</p> <p>1 = Derivative on a stock</p> <p>2 = Derivative on a derivative. E.g., an option on a future of a stock.</p>
dec_in_strike_price_n	<p>uint16_t</p> <p>Number of implicit decimals in the strike_price_i field (part of series_t structure). A value of zero means no decimals.</p>
dec_in_contr_size_n	<p>uint16_t</p> <p>Number of implicit decimals in the contract_size_i and the price_quot_factor_i fields. A value of zero means no decimals.</p>
rnt_id_n	<p>uint16_t</p> <p>This identifies how the instrument is ranked. Always set to one.</p>
virt_commodity_n	<p>uint16_t</p> <p>When distributing broadcasts classified with information type "Instrument Class", a virtual underlying can be used to group a number of instrument classes together. The virtual underlying can be used in broadcast subscriptions.</p> <p>If zero, no virtual underlying is used but the real underlying code is used in broadcast subscriptions.</p>
settlement_days_n	<p>uint16_t</p>

Variable	Description
	Number of settlement days or months used in the calculation rule. Refer to the settl_day_unit_c field.
settl_day_unit_c	uint8_t Describes the unit of the number of Settlement Days Rule for the instrument class. Possible values: 0 = Not applicable 1 = Days 2 = Months.
inc_id_s	char[14] The short ASCII representation of the instrument class.
name_s	char[32] The full ASCII representation of the instrument class.
trc_id_s	char[10] The ID string for a trade report class. The trade report class contains a list of Trade Report Types.
base_cur_s	char[3] A three-letter currency identifier, e.g. AUD indicating Australian dollars. Indices, i.e. instrument classes with a price_unit_premium_c set to points, will still have a value in this field.
traded_c	uint8_t Indicates if the corresponding series for the instrument class are traded. Possible values: 1 = Yes 2 = No.
price_unit_premium_c	uint8_t The premium unit for the class. Possible values: 1 = Price 2 = Yield 3 = Points 4 = Yield difference 5 = IMM index 6 = Basis points 7 = Inverted yield 8 = Percentage of nominal 9 = Dirty Price.
price_unit_strike_c	uint8_t The strike price unit for the class. Possible values: 1 = Price 2 = Yield 3 = Points 6 = Basis points 7 = Inverted yield.
indicative_prices_c	uint8_t Indicative prices. Possible values: 1 = Yes 2 = No.
trd_cur_unit_c	uint8_t

Variable	Description
	<p>Specifies the currency unit the instrument is traded in. Possible values:</p> <ul style="list-style-type: none"> 1 = Primary unit 2 = Secondary unit 3 = Tertiary unit. <p>ASX Trade allows each currency up to three units of measurement. E.g., Australian dollars has a primary unit of “dollar”, a secondary unit of “cent”, and no tertiary unit.</p>
db_operation_c	<p>uint8_t</p> <p>Operation that the central system has carried out on this particular item.</p> <p>Note: Users are required to store in their OI applications a copy of the instrument classes that concern them. An insert could appear for an item that they have already recorded or an update could appear for an item that they have yet to record. These situations need to be catered for by the user. As mentioned above, a remove will be indicated by the named structure ns_remove_t (named structure 37002).</p> <p>Possible values:</p> <ul style="list-style-type: none"> 1 = Insert 2 = Update.
csd_id_s	<p>char[12]</p> <p>Ignore. Not used by ASX Trade.</p>
filler_2_s	<p>char[2]</p> <p>Ignore. Used for byte alignment.</p>

23.3.8 ns_price_tick_t (named structure 37102)

For each class there can be several tick size structures provided. However, the way ASX Trade is currently configured, there is allowance for only one value for the entire class for the decimals in premium. Therefore, for every tick size of a certain instrument class, the dec_in_premium_n value will be the same.

Variable	Description
tick_size	<p>tick_size_t</p> <p>See tick_size_t sub structure below.</p>
dec_in_premium_n	<p>uint16_t</p> <p>Number of implicit decimals in the premium/price.</p>
is_fractions_c	<p>char[1]</p> <p>Indicates if the premium is represented as fractions. Always set to ‘N’ meaning “No.”</p>
price_format_c	<p>uint8_t</p> <p>Premium/price format. Always set to 0.</p>

23.3.9 tick_size_t

Variable	Description
step_size_i	<p>int32_t</p> <p>The tick size in the minimum valid step in the premium or price.</p>
lower_limit_i	<p>int32_t</p> <p>The lower limit in an interval where the actual tick size is valid.</p>
upper_limit_i	<p>int32_t</p> <p>The upper limit in an interval where the actual tick size is valid.</p>

23.3.10 ns_block_size_t (named structure 37103)

Variable	Description
maximum_size_u	int64_t The maximum volume allowed for the order per block size. Currently set to 2,147,483,647 for all instrument classes.
minimum_size_n	uint32_t The minimum volume required for the order per block size. Currently set to 0 for all instrument classes, indicating no minimum volume.
block_n	uint32_t Defines the block size of the class. Currently set to 1 for all instrument classes.
lot_type_c	uint8_t Type of lot. Possible values: 1 = Odd lot 2 = Round lot 3 = Block lot 4 = All or none. Currently set to 2 for all instrument classes.
filler_3_s	char[3] Ignore. Used for byte alignment.

23.3.11 ns_calc_rule_t (named structure 37104)

Variable	Description
accr_intr_round_u	uint32_t Accrued interest rounding.
clean_pr_round_u	uint32_t Clean price rounding.
yield_conv_n	uint16_t Yield convention. Number of months.
ex_coupon_n	uint16_t Ex coupon period.
accr_intr_ud_c	uint8_t Accrued interest up/down. Possible values: 1 = Up 2 = Down.
clean_pr_ud_c	uint8_t Clean price up/down. Possible values: 1 = Up 2 = Down.
day_count_conv_c	uint8_t Day count convention. Possible values: 1 = ACTACT 2 = ACTAFB 3 = EU30360 4 = US30360

Variable	Description
	5 = ACT365 6 = ACT360.
eom_count_conv_c	uint8_t End of month count convention. Possible values: 1 = SAME 2 = LAST360 3 = LAST.
set_start_consider_c	uint8_t Start consideration. Possible values: 1 = Yes 2 = No.
set_end_consider_c	uint8_t End consideration. Possible values: 1 = Yes 2 = No.
calculation_conv_c	uint8_t Calculation convention. Possible values: 1 = Compound 2 = Compound Simple 3 = Simple_MM 4 = Discount 5 = US Treasury 6 = Proceed.
cadj_trade_price_c	uint8_t Ignore. Not used by ASX Trade.
ex_coupon_calc_type_c	uint8_t Ignore. Not used by ASX Trade.
filler_3_s	char[3] Ignore. Used for byte alignment.

23.3.12 ns_crossing_rule_t (named structure 37106)

A set of crossing rules can be assigned to a particular instrument class. Note that this structure will only appear in the broadcast if the crossing rule from the associated instrument type is not applicable for this class. For a brief discussion of the rules on crossing refer to *ASX Trade Introduction and Business Information*.

ASX Trade has implemented crossing rules at the instrument type level and it is unlikely that these will be overridden at the instrument class level in future. This structure is given here in case such an event were to occur.

Variable	Description
min_wait_time_i	int32_t This is the minimum amount of time, specified in seconds, for an order to have existed in the order book before a cross-with-book can take place. If the existing order was deleted so that a two-sided crossing could be made then the balance of the order would lose market priority.
max_wait_time_i	int32_t

Variable	Description
	Relates to a two-sided crossing and a cross-with-book in a derivatives market. This is the time, specified in seconds, that a user must wait before proceeding to attempt the crossing after issuing a crossing quote request, i.e., how long a user must wait regardless if a market is established or not before they can cross.
max_time_span_i	int32_t This is the maximum time, specified in seconds, that a user is given after waiting the required period for responses to the crossing quote request to complete a derivatives crossing. Whether a market has been established or not for a crossing order, the user must complete the crossing within this time frame.
min_hold_time_i	int32_t This is the minimum time, specified in seconds, that a crossing order must be exposed to the market. For a two-sided crossing, this time has to lapse for each side, beginning with the bid, then the ask. I.e. this is the least amount of time the participant is allowing the market to match the order.
max_time_illegal_i	int32_t This is the maximum time, specified in seconds, that a crossing order can be examined to determine if it was illegal or not. Once this time has lapsed, the crossing order cannot be challenged.
mar_min_time_i	int32_t For priority and non-priority crossings, this field specifies the minimum time in seconds that a crossing market has to have been established. Once this time has lapsed the user can execute their crossing order.
mar_max_spread_i	int32_t For priority crossings, a crossing market must exist. This is where the best offer and the best ask are within a certain price tick spread. This field determines the maximum of this spread in price ticks.
max_time_cp_priority_i	int32_t Maximum time, centre point priority crossing in seconds.
allow_priority_c	uint8_t Specifies if priority crossing is allowed or not. Possible values: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.

23.3.13 ns_inst_class_ext1_t (named structure 37107)

Variable	Description
initial_trr_min_value_u	int64_t The required minimum trade value for initial trade reports.
ext_info_s	char[32] Additional extended information for the instrument class.
non_chess_c	uint8_t Non CHESSE settlement. Possible values: 1 = Yes 2 = No.

Variable	Description
for_flag_c	uint8_t Foreign ownership restriction. Incorporated outside country. Possible values: 1 = Yes 2 = No.
practice_sec_c	uint8_t The instrument is used for training and practice purposes. Possible values: 1 = Yes 2 = No.
filler_1_s	char[1] Ignore. Used for byte alignment.

24 BU124 Delta Instrument Series Update

24.1 Broadcast Function

This broadcast is used to send out information about a new instrument series or an instrument series that has changed.

Since this broadcast has an information type of “General” its subscription cannot be filtered nor can ASX tailor the configuration of its dissemination. Therefore subscribed participants will receive broadcasts on instruments to which they have no access. The OI application must handle this situation.



Note:

ASX Trading Operations current practice is to add, delete and amend instrument series overnight and not during the trading day. However they can create a special market intra-day and hence use this broadcast.

24.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	This is a VIB and has no top most structure. Instead, the sequence of possible structs are described in the <i>Message Structure</i> section below.
Information Type	General

24.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_segment_hdr_t
- one of:
 - item_hdr_t
 - one of:
 - sub_item_hdr_t
 - a choice either of:
 - ns_delta_hdr_t (named structure 37001)
- one or more sequences of:
 - item_hdr_t
 - one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - ns_remove_t (named structure 37002)
 - or both the following:
 - ns_inst_series_basic_t (named structure 37301)
 - ns_inst_series_basic_single_t (named structure 37302).

24.3.1 broadcast_segment_hdr_t

Currently in ASX Trade, out of all the broadcasts that contain this header, only the BU120 will use the segment_number_n field. It indicates that information relating to one entity has been split over two or more broadcasts. The header is provided in this broadcast in case it is required in the future.

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 124}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the structure that is to follow, including this header.
segment_number_n	uint16_t Ignore. Not used in this broadcast.
filler_3_s	char[2] Ignore. Used for byte alignment.

24.3.2 item_hdr_t

Variable	Description
items_n	uint16_t The number of sub-items following this item header.
size_n	uint16_t The size of the structure that is to follow, including this header.

24.3.3 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 The size of the structure that is to follow, including this header.

24.3.4 ns_delta_hdr_t (named structure 37001)

Variable	Description
download_ref_number_q	int64_t Reference number to use in delta queries and answers. This number is always increasing but may contain gaps. The application needs to store this number so that it may use it again for the next time it makes a DQ124 query, and thereby only retrieve the delta information.
full_answer_timestamp	time_spec_t See time_spec_t sub structure below.
full_answer_c	uint8_t A full answer is enforced in the delta query. Possible values: 1 = Yes 2 = No.

Variable	Description
filler_3_s	char[3] Ignore. Used for byte alignment.

24.3.5 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since 1970-01-01 00:00:00 UTC.
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

24.3.6 ns_remove_t (named structure 37002)

Indicates that the series has been removed. Only this structure is sent if the series has been removed.



Note:

Participants are required to store in their OI applications a copy of the series that concern them. A remove structure could appear for an item that is unknown to them. Participants need to cater for this situation.

Variable	Description
series_t	series_t Contains the series that has been removed.

24.3.7 ns_inst_series_basic_t (named structure 37301)

Both this structure and ns_inst_series_basic_single_t (named structure 37302) are sent if a series has been added or modified.

Variable	Description
series	series_t Contains the series to which this information relates.
step_size_multiple_n	uint16_t Tick size multiple is used to calculate the tick size for the instrument. The tick size itself is distributed in the instrument class. If the same tick size is used for all instrument series belonging to the class, the value in this field will be 1 for all instruments.
settlement_days_n	uint16_t Number of settlement days or months used in the calculation rule. Refer to the settl_day_unit_c field.
ins_id_s	char[32] The unique series name in ASCII characters.
long_ins_id_s	char[32] Additional instrument series information in ASCII characters. May be blank and may not necessarily be unique across all series.
date_last_trading_s	char[8]

Variable	Description
	The last valid trading date of the series in YYYYMMDD format. The date here, together with the time_last_trading_s field, provides the last trading time in Universal Time Code (UTC). For equities, this field is set to all blanks.
time_last_trading_s	char[6] The last valid trading time of the series in HHMMSS format. The time here, together with date_last_trading_s field provides the last trading time in UTC. For equities, this field is set to all blanks.
date_first_trading_s	char[8] The first valid trading date of the series in YYYYMMDD format. This date here, together with the time_first_trading_s field provides the first trading time in UTC.
time_first_trading_s	char[6] The first valid trading time of the series in HHMMSS format. This time field, together with the date_first_trading_s field provides the first trading time in UTC.
series_status_c	uint8_t The actual state of the series. Possible values: 1 = Active (both expired and not expired) 2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status) 3 = Issued (series exists in the system but is unavailable for trading – to the participant this is the same as delisted) 4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values: 1 = Yes 2 = No.
db_operation_c	uint8_t Operation that the central system has carried out on this particular item. Note: Participants are required to store in their OI applications a copy of the series that concern them. An insert could appear for an item that they have already recorded or an update could appear for an item that they have yet to record. These situations need to be catered for by the participant. As mentioned previously, a remove will be indicated by the named structure ns_remove_t (named structure 37002). Possible values: 1 = Insert 2 = Update.
trade_reporting_only_c	uint8_t Ignore, not used. Always set to 2.
settl_day_unit_c	Describes the unit of the number of Settlement Days Rule for the instrument series. Possible values: 0 = Not applicable - Settlement Days Rule set at Instrument Class is used 1 = Days

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

24.3.8 ns_inst_series_basic_single_t (named structure 37302)

Both this structure and ns_inst_series_basic (named structure 37301) are sent if a series has been added or modified.

Variable	Description
upper_level_series	series_t This field only contains information if the series to which this whole structure relates to is a second level derivative. I.e., if the series is a derivative of a derivative. Derivative levels for a series are defined in its associated instrument class. There is no guarantee when retrieving a full answer that an instrument series will be received by the participant before any others that are linked to it through this field.
contract_size_i	int32_t For an option, this indicates the contract size, i.e., the number of units of the underlying security that concerns one unit of the instrument. For example, buying 10 call options with a contract size of 100 will give the buyer the right to buy 10 * 100 = 1,000 of the underlying security. For a warrant the contract size indicates the number of units of the warrant that concerns one unit of the underlying security. E.g., a value of four indicates that four warrants are equivalent to one underlying security. The exceptions to this rule are for index warrants where it represents the cash value per index point, and for basket warrants where it represents the minimum parcel of warrants required to be held for early exercise. This field only has a value if it differs from the associated instrument class.
price_quot_factor_i	int32_t The factor is used to calculate the total price for a trade. This will only have a value if it differs from the associated instrument class.
state_number_n	uint16_t Ignore, not used.
ex_coupon_n	uint16_t Ex coupon period.
isin_code_s	char[12] A code that uniquely identifies a specific securities issue (ISIN). The ISIN consists of: <ul style="list-style-type: none"> • A prefix, which is the alpha-2 country code • The basic code, which is nine characters • A check digit. For more information about ISIN code, see the international standard ISO 3166.
settlement_date_s	char[8] Ignore. Not used.
first_settlement_date_s	char[8] First settlement date in YYYYMMDD format.
date_notation_s	char[8]

Variable	Description
	Notation date in YYYYMMDD format.
deliverable_c	uint8_t Defines if a series can be delivered or not (cash settlement). Possible values: 1 = Yes 2 = No.
effective_exp_date_s	char[8] The effective expiration date is the actual expiration date of the series and is normally the same as expiration_date_n in the series binary code. The effective expiration date can be changed during the lifetime of the series, whereas expiration_date_n continues to hold the original expiration date. Format: YYYYMMDD.
ext_info_source_c	uint8_t Specifies whether or not the data source for distributed prices is sent into the system with an external transaction. Possible values: 1= Yes 2 = No.
filler_2_s	char[2] Ignore. Used for byte alignment.

25 BU126 Combination Series Update

25.1 Broadcast Function

This broadcast is used to send out information about a new combination series or a combination series that has been changed.

25.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	This is a VIB and has no topmost structure. Instead, the sequence of possible structs are described in the Message Structure section below.
Information Type	General.

25.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_segment_hdr_t
- one or more sequences of:
 - item_hdr_t
 - one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - ns_inst_series_basic_t (named structure 37301)
 - ns_combo_series_leg_t (named structure 37308).

25.3.1 broadcast_segment_hdr_t

Currently in ASX Trade, out of all the broadcasts that contain this header, only the BU120 uses the segment_number_n field. It indicates that information relating to one entity has been split over two or more broadcasts. The header is provided in this broadcast in case it is required in the future.

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'B', 'U', 126}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the structure that is to follow, including this header.
segment_number_n	uint16_t Ignore. Not used in this broadcast.
filler_2_s	char[2] Ignore. Used for byte alignment.

25.3.2 item_hdr_t

Variable	Description
items_n	uint16_t The number of sub-items following this item header.
size_n	uint16_t The size of the structure that is to follow, including this header.

25.3.3 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 The size of the structure that is to follow, including this header.

25.3.4 ns_inst_series_basic_t (named structure 37301)

Variable	Description
series	series_t Contains the series to which this information relates.
step_size_multiple_n	uint16_t Tick size multiple is used to calculate the tick size for the instrument. The tick size itself is distributed in the instrument class. If the same tick size is used for all instrument series belonging to the class, the value in this field is one for all instruments.
settlement_days_n	uint16_t Number of settlement days or months used in the calculation rule. Refer to the settl_day_unit_c field.
ins_id_s	char[32] The unique series name in ASCII characters.
long_ins_id_s	char[32] Additional instrument series information in ASCII characters. May be blank and may not necessarily be unique across all series.
date_last_trading_s	char[8] The last valid trading date of the series in YYYYMMDD format. The date here, together with the time_last_trading_s field provides the last trading time in UTC. For equities, this field is set to all blanks.
time_last_trading_s	char[6] The last valid trading time of the series in HHMMSS format. The time here, together with date_last_trading_s field provides the last trading time in UTC. For equities, this field is set to all blanks.
date_first_trading_s	char[8] The first valid trading date of the series is in YYYYMMDD format. This date here, together with the time_first_trading_s field provides the first trading time in UTC.
time_first_trading_s	char[6] The first valid trading time of the series in HHMMSS format. This time field, together with the date_first_trading_s field provides the first trading time in UTC.

Variable	Description
series_status_c	uint8_t The actual state of the series. Possible values: 1 = Active (both expired and not expired) 2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status). 3 = Issued (series exists in the system but is unavailable for trading – to the participant this is the same as delisted) 4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values: 1 = Yes 2 = No.
db_operation_c	uint8_t Operation that the central system has carried out on this particular item. Note: Participants are required to store in their OI applications a copy of the series that concern them. An insert could appear for an item that they have already recorded or an update could appear for an item that they have yet to record. These situations need to be catered for by the participant. As mentioned previously, a remove is indicated by the named structure ns_remove_t (named structure 37002). Possible values: 1 = Insert 2 = Update.
trade_reporting_only_c	uint8_t Ignore, not used. Always set to 2.
settl_day_unit_c	Describes the unit of the number of Settlement Days Rule for the instrument series. Possible values: 0 = Not applicable - Settlement Days Rule set at Instrument Class is used 1 = Days 2 = Months.
filler_2_s	char[2] Ignore. Used for byte alignment.

25.3.5 ns_combo_series_leg_t (named structure 37308)

Variable	Description
series	series_t The series identifying this particular leg of the combination.
ratio_n	uint16_t Relative numbers of contracts in comparison to other legs.
op_if_buy_c	char[1] Specifies whether to buy or sell the series when buying the combination. Possible values:

Variable	Description
	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.

26 CB15 Directed Trade Broadcast

26.1 Broadcast Function

The CB15 is a directed trade broadcast that:

- is sent to the participant that owns the trade (and a BSP if requested by the participant) upon trade execution
- contains all trade details, including public (e.g. quantity, price) and restricted information (e.g. client reference, order condition)
- removes the requirement to download full order history at login therefore enabling OI applications to be more efficient
- carries the information about a single side of a trade, that is, only the bid side or the ask side, and not both.

26.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead it holds a sequence of possible structs that are described below.
Information Type	Instrument Dedicated.

26.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_hdr_t
- one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - cl_trade_base_api_t (named structure 3)
 - cl_trade_asx_api_t (named structure 65)
 - cl_deal_extended_price_asx_t (named structure 34)
 - cl_deal_trade_report_asx_t (named structure 26)
 - cl_combo_series_asx_t (named structure 42)
 - cl_trade_regulatory_asx_t (named structure 35)
 - cl_trade_short_asx_t (named structure 36).

26.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'C', 'B', 15}.
items_n	uint16_t The number of sub-items held in this structure.
size_n	uint16_t The total size of the message, including this header.

26.3.2 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 The size of the structure that is to follow, including this header.

26.3.3 cl_trade_base_api_t (named structure 3)

Variable	Description
trading_code	trading_code_t Contains the participant trading code to which this side of the trade belongs. See trading_code_t sub structure below.
series	series_t Contains the series to which this trade information relates.
give_up_member	give_up_member_t See give_up_member_t sub structure below.
order_number_u	quad_word Identifies the order taking part in this side of the trade.
sequence_number_i	int32_t A unique number per day per participant that can be used to synchronise the broadcast with the associated query.
trade_number_i	int32_t A number that can be used to relate information in this broadcast with the associated information in a CB16.
deal_price_i	int32_t The price at which the trade occurred.
trade_quantity_i	int64_t Defines the traded quantity.
account	account_t See account_t sub structure below.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to the participant their own order identifier.
bought_or_sold_c	uint8_t Defines if this order indicated by the order_number_u field was on the bid or ask side of the market. Possible values: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the deal is created. Refer to <i>ASX Trade Introduction and Business Information - Trade Source</i> for the possible values in this field.
open_close_req_c	uint8_t Not used. Possible Values: 0 = Default

Variable	Description
	1 = Open 2 = Close/net 3 = Mandatory close 4 = Set to default to the account (valid only for alter order)
trade_type_c	uint8_t Defines the type of trade. Possible values: 1 = Standard - The trade is a normally a registered trade. 4 = Reversing - The trade has been cancelled.
le_state_c	uint8_t Ignore. Not used by ASX Trade. Value will always be set to four.
user_code	user_code_t Specifies a recipient of the message. An item in the list can either be a participant (exchange code and customer code) or an individual user (exchange code, customer code, and user signature). See user_code_t sub structure below.
created_date_s	char[8] The date of the trade, quoted in local time. Format YYYYMMDD.
created_time_s	char[6] The time of trade creation. Format: HHMMSS.
asof_date_s	char[8] For ASX Trade the value in this field is the same as created_date_s.
asof_time_s	char[6] The time of matching.
modified_date_s	char[8] The date the item was last changed. Format: YYYYMMDD.
modified_time_s	char[6] The time the item was last changed. Format: HHMMSS.
trade_state_c	uint8_t The state of the trade. Always set to 1 (active).
attention_c	uint8_t Ignore. Not used.
deal_number_i	int32_t A number that identifies a specific deal. Deal number is unique within instance and instrument type.
global_deal_no_u	uint32_t A number that together with series identifies a specific deal.
orig_trade_number_i	int32_t Ignore. Not used by ASX Trade.
orig_series	orig_series_t Same structure as series_t. Only used for trade cancellations.

Variable	Description
exchange_info_s	char[32] A free text field used at participant's discretion. For trade reports, 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.
big_attention_u	uint32_t This field is a bitmask providing further information on the trade. The default value is: 0 = Ignore The following bits may be set: 32 [bit 6] = Aggressive Order – the trade is the part created by an incoming order as opposed to the part that was already stored in the order book. 2048 [bit 12] = Short Sell 2097152 [bit 22] = Delayed trade report resulting from ITN 4194304 [bit 23] = Practice Security
clearing_date_s	char[8] The date of clearing in YYYYMMDD format.
execution_timestamp	time_spec_t Time the trade was entered into the system or execution time for on market trades. See time_spec_t sub structure.
trade_venue_c	uint8_t Defines the Trade venue, i.e. from where the trade emanates. Always set to one.
instance_c	uint8_t Denotes the instance number of the Trade Handler process. Set to one
exch_order_type_n	uint16_t Exchange specific order types. Ignore any values returned that are not in the list below: 2 = Short Sell order: (premium_i = an integer, order_type_c = 1 = Limit order premium_i = 0, order_type_c = 2 = Market order premium_i = 0, order_type_c = 3 Market-to-Limit order premium_i = 0, order_type_c = 17, time_validity_n != 0 = Best-Limit order). Can also be combined with other exchange specific order types/attributes outlined below. Can only be a sell order (bid_or_ask_c = 2). 4 = Market Bid order: (premium_i = an integer, order_type_c = 1, only entered by ASX Trading Operations). 8 = Price Stabilisation/Green Shoe Order: (premium_i = an integer, order_type_c = 1, time_validity_n != 0). 32 = Undisclosed (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 limit or market-to-limit) 4096 = Centre Point Block order (use order_type_c to determine market or limit).
party	party_t This is the declared counter party for the other side of the trade. See party_t sub structure.
trade_rep_code_n	uint16_t

Variable	Description
	Ignore. Not used by ASX Trade.
filler_2_s	char[2] Ignore. Used for byte alignment.
match_id	match_id_t Match ID represents a unique combination of execution event number, execution group number and match item number.

26.3.4 give_up_member_t

This structure defines the clearing participant information for the trade.

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 participant 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.

26.3.5 account_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.
account_id_s	char[10] The account identifier.
filler_3_s	char[3] Ignore. Used for byte alignment.

26.3.6 user_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.

Variable	Description
user_id_s	char[5] The unique identifier of the participant who sent the message. Participants can retrieve their own identifier using the omniapi_get_info_ex() function.

26.3.7 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.
initial_trd_report_c	uint8_t Indicates if the trade report is the initial part of an Initial or Delayed Trade Report. Possible values: 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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.

26.3.8 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since the Epoch (1970-01-01 00:00:00 UTC).
tv_nsec	int32_t Elapsed time in nanoseconds since the seconds in tv_sec.

26.3.9 cl_trade_asx_api_t (named structure 65)

Variable	Description
party	party_t This is the declared counterparty for the other side of the trade. See party_t sub structure below.
settlement_date_s	char[8] The settlement date for the trade. Format: YYYYMMDD.
trade_slip_number_i	int32_t ASX Trade Slip number. Of the format 1OPNNNNNNN: 1 - always 1

Variable	Description
	O - last digit of the Ordinal date P - Instance number of DC NNNNNNN - Trade number.
trade_condition_n	uint16_t The condition in which a trade was executed. Possible values: 0 = No Condition 2 = Internal Trade/Crossing 8 = Buy Write (Equity/Derivative Combination). This field acts as a bit mask. The binary AND operator can be used on the above possible values. Refer to ASX Trade Markets, Instrument Groups and Trade Condition Codes .
ext_t_state_c	uint8_t Trade report codes. Possible value: 0 = Ignore, not relevant. For more information refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> .
counter_order_capacity_c	char[1] Dealing capacity of the counter order for crossings. For crossed trades, this returns the value provided in capacity_of_participant_s of the opposing order. For noncrossed trades, zero will be returned.

26.3.10 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 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.
filler_1_s	char[1] Ignore. Used for byte alignment.

26.3.11 cl_deal_extended_price_asx_t (named structure 34)

This is not an error; this is the same structure as used in cl_deal_trade_report_asx_t. It is used for the purpose of disseminating trades from Centre Point orders where greater precision is required in the traded price.

Variable	Description
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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.
as_of_date_s	char[8] The date of the reported trade, quoted in UTC time.

Variable	Description
	Format YYYYMMDD.
item	cl_deal_extended_price_asx_item_t[3] There are always three items in the array. See cl_deal_extended_price_asx_item_t substructure below.

26.3.12 cl_deal_extended_price_asx_item_t

Variable	Description
corp_action_code_s	char[2] Corporate action code. Can be NULL indicating no entry for this item in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.

26.3.13 cl_deal_trade_report_asx_t (named structure 26)

Variable	Description
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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.
as_of_date_s	char[8] The date of the reported trade. Format YYYYMMDD.
item	cl_deal_trade_report_asx_item_t[3] There are always three items in the array. See cl_deal_trade_report_asx_item_t sub structure below.

26.3.14 cl_deal_trade_report_asx_item_t

Variable	Description
corp_action_code_s	char[2] Corporate action code. Can be NULL indicating no entry for this item in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.

26.3.15 cl_combo_series_asx_t (named structure 42)

This structure appears in the CB15 for the equity leg only, when the trade involves a TMC and that combination includes a mixture of equity and derivative type series. It is not included in the broadcast if the TMC contained only equities, or contained only derivatives.

Variable	Description
new_series	new_series_t Same structure as series_t. Indicates the derivative series that is included in the TMC trade.

26.3.16 cl_trade_regulatory_asx_t (named structure 35)

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.

26.3.16.1 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.

26.3.17 cl_trade_short_asx_t (named structure 36)

Variable	Description
trade_short_sell_quantity_i	int64_t Quantity of the trade that is short (partial or whole). Only applies to short sell trades.

27 CB16 Trade

27.1 Broadcast Function

This broadcast is used for disseminating information on trades made in the market. One broadcast contains information about one trade. The only time that this is not the case is when a combination that includes an equity leg plus derivative leg(s) trades. In this case the trade information of the equity leg is contained in a separate CB16 broadcast that immediately follows the CB16 of the first derivative leg.

27.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead it holds a sequence of possible structs that are described below.
Information Type	General
Segmented	True

27.3 Message Structure

This is a variable information message. Headers and sub headers within the message identify what is contained in the message. The overall structure is:

- broadcast_hdr_t
- one or more sequences of:
 - sub_item_hdr_t
 - a choice of:
 - cl_deal_base_t (named structure 24)
 - cl_deal_asx_t (named structure 25)
 - cl_trade_public_asx_t (named structure 27)
 - cl_trade_anonymous_asx_t (named structure 28)
 - cl_deal_trade_report_asx_t (named structure 26)
 - cl_trade_cancel_asx_t (named structure 30)
 - cl_deal_extended_price_asx_t (named structure 34)
 - cl_combo_series_asx_t (named structure 42).

A cl_deal_base_t structure is always the leading constituent in the variable sequence of structures.

For equity and interest rate markets the CB16 broadcasts will disseminate anonymous deals information. For derivatives markets the broadcast is generated with counterparty information.

There are several possible compositions of the CB16, depending on the type of trade that it is broadcasting. The following indicates what the broadcast would contain given the particular deal in the market:

- A non-anonymous regular trade:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_trade_public_asx_t
 - cl_trade_public_asx_t
- An anonymous regular trade:
 - cl_deal_base_t

- cl_deal_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_anonymous_asx_t
- A non-anonymous trade report:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_deal_trade_report_asx_t
 - cl_trade_public_asx_t
 - cl_trade_public_asx_t
- An anonymous trade report:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_deal_trade_report_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_anonymous_asx_t
- A non-anonymous regular trade cancellation:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_trade_public_asx_t
 - cl_trade_cancel_asx_t
 - cl_trade_public_asx_t
 - cl_trade_cancel_asx_t
- An anonymous regular trade cancellation:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_cancel_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_cancel_asx_t
- A non-anonymous trade report cancellation:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_deal_trade_report_asx_t
 - cl_trade_public_asx_t
 - cl_trade_cancel_asx_t
 - cl_trade_public_asx_t
 - cl_trade_cancel_asx_t
- An anonymous trade report cancellation:
 - cl_deal_base_t
 - cl_deal_asx_t
 - cl_deal_trade_report_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_cancel_asx_t
 - cl_trade_anonymous_asx_t
 - cl_trade_cancel_asx_t.

27.3.1 broadcast_hdr_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'C', 'B', 16}.
items_n	uint16_t The number of sub-items held in this structure.
size_n	uint16_t The total size of the message, including this header.

27.3.2 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 The size of the structure that is to follow, including this header.

27.3.3 cl_deal_base_t (named structure 24)

Variable	Description
series	series_t Contains the series to which this trade information relates.
execution_timestamp	time_spec_t Time the trade was entered into the system, or execution time for on market trades. See time_spec_t sub structure.
sequence_number_u	uint32_t The sequence number assigned to this particular broadcast. These sequence numbers start at one per day. Participants may not be permitted to access some trades and therefore encounter gaps in numbering.
deal_price_i	int32_t The price at which this trade was executed.
deal_quantity_i	int64_t The quantity of the trade.
segment_number_n	uint16_t Ignore. Not used in this case.
instance_c	uint8_t Denotes the instance number of the Trade Handler process. Set to one
filler_1_s	char[1] Ignore. Used for byte alignment.

27.3.4 time_spec_t

Variable	Description
tv_sec	uint32_t Elapsed time in seconds since the epoch (1970-01-01 00:00:00 UTC).
tv_nsec	int32_t

Variable	Description
	Elapsed time in nanoseconds since the seconds in tv_sec.

27.3.5 cl_deal_asx_t (named structure 25)

Variable	Description
trade_slip_number_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.
exch_order_type_n	uint16_t Exchange specific order types. 4 = Market Bid order (only entered by ASX Trading Operations) 8 = Price Stabilisation/Green Shoe Order 64 = Centre Point Order (use order_type_c to determine market or limit).
trade_condition_n	uint16_t The condition in which a trade was executed. Possible values: 0 = No Condition 2 = Internal Trade/Crossing 8 = Buy Write (Equity/Derivative Combination). This field acts as a bit mask. The binary AND operator can be used on the above possible values. Refer to ASX Trade Markets, Instrument Groups and Trade Condition Codes for more information.
settlement_date_s	char[8] Settlement date in the format YYYYMMDD.
ext_t_state_c	uint8_t Trade report codes. Possible value: 0 = Ignore, not relevant. Refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> .
filler_3_s	char[3] Ignore. Used for byte alignment.

27.3.6 cl_trade_public_asx_t (named structure 27)

Variable	Description
participant	participant_t See participant_t sub structure below.
order_number_u	quad_word Identifies the order taking part in this side of the trade.
block_n	uint32_t Ignore. Set to one or zero.
bought_or_sold_c	uint8_t Defines if this order indicated by the order_number_u field was on the bid or ask side of the market. Possible values: 1 = Bought

Variable	Description
	2 = Sold.
deal_source_c	uint8_t Where the deal was created. Refer to <i>ASX Trade Introduction and Business Information - Trade Source</i> for more information.
filler_2_s	char[2] Ignore. Used for byte alignment.
trade_number_i	int32_t A number that can be used to relate information in this broadcast with the associated information in a CB15.
big_attention_u	uint32_t This field is a bitmask providing further information on the trade. The default value is: 0 = Ignore The following bits may be set: 32 [bit 6] = Aggressive Order – the trade is the part created by an incoming order as opposed to the part that was already stored in the order book. 2048 [bit 12] = Short Sell 4194304 [bit 23] = Practice Security

27.3.7 participant_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.

27.3.8 cl_trade_anonymous_asx_t (named structure 28)

Variable	Description
order_number_u	quad_word Identifies the order taking part in this side of the trade.
block_n	uint32_t Ignore. Set to one or zero.
bought_or_sold_c	uint8_t Defines if this order, indicated by the order_number_u field, was on the bid or ask side of the market. Possible values: 1 = Bought 2 = Sold.
deal_source_c	uint8_t

Variable	Description
	Where the deal was created. Refer to <i>ASX Trade Introduction and Business Information - Trade Source</i> for the possible values in this field.
filler_2_s	char[2] Ignore. Used for byte alignment.
trade_number_i	int32_t A number that can be used to relate information in this broadcast with the associated information in a CB15.
big_attention_u	uint32_t This field is a bitmask providing further information on the trade. The default value is: 0 = Ignore The following bits may be set: 32 [bit 6] = Aggressive Order – the trade is the part created by an incoming order as opposed to the part that was already stored in the order book. 2048 [bit 12] = Short Sell 4194304 [bit 23] = Practice Security

27.3.9 cl_deal_trade_report_asx_t (named structure 26)

Please note that this structure will not be disseminated for accounting trade types (for example, Booking Purpose trades).

Variable	Description
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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.
as_of_date_s	char[8] The date of the reported trade. Format: YYYYMMDD.
item	cl_deal_trade_report_asx_item_t[3] There are always three items in the array. See cl_deal_trade_report_asx_item_t sub structure below.

27.3.10 cl_deal_trade_report_asx_item_t

Variable	Description
corp_action_code_s	char[2] Corporate action code. Can be NULL indicating no entry for this item in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.

27.3.11 cl_trade_cancel_asx_t (named structure 30)

Variable	Description
as_of_date_s	char[8] The date that the trade cancellation occurred. Format: YYYYMMDD.
as_of_time_s	char[6] The time that the trade cancellation occurred. Format: HHMMSS.
modified_date_s	char[8] The date the item was last changed. Format: YYYYMMDD.
modified_time_s	char[6] The time the item was last changed. Format: HHMMSS.
trade_type_c	uint8_t Always set to four, indicating a cancelled trade.
filler_3_s	char[3] Ignore. Used for byte alignment.

27.3.12 cl_combo_series_asx_t

This structure will appear in the CB16 when the trade involves a TMC and that combination includes a mixture of equity and derivative type series. It will not be included in the broadcast if the TMC contained only equities, or contained only derivatives. It is only needed at recovery when a derivatives user executes a CQ27 query and needs to determine the equity trades of a TMC from other equity trades (refer to *CQ27 Missing Deals* in *ASX Trade Queries*).

Variable	Description
new_series	new_series_t Same structure as series_t. It indicates the derivative series that is included in the TMC trade.

27.3.13 cl_deal_extended_price_asx_t (named structure 34)

Variable	Description
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 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.
as_of_date_s	char[8] This field is blank and to be ignored within this struct.
item	cl_deal_extended_price_asx_item_t[3] There are always three items in the array. See cl_deal_extended_price_asx_item_t substructure below.

27.3.14 cl_deal_extended_price_asx_item_t

Variable	Description
corp_action_code_s	char[2] Corporate action code. Can be NULL indicating no entry for this item in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.

28 MI4 Quote Request with Volume Information

28.1 Broadcast Function

This broadcast is sent after a valid Quote Request with Volume (refer to *MC4 Quote Request with Volume* in *ASX Trade Transactions*). The broadcast is sent when the Quote Request is supposed to be sent to the entire market.

The responsible market maker as well as other users may respond to this by sending in orders.

28.2 Broadcast Properties

Function Call	omniapi_read_event_ext_ex or omniapi_read_event_block
Struct Name	quote_request_vol_info_t
Information Type	Derivative

28.3 Message Structure

28.3.1 quote_request_vol_info_t

Variable	Description
broadcast_type	broadcast_type_t Contains the following: {'M', 'I', 4}.
series	series_t The series for which the broadcast was issued.
user_code	user_code_t Specifies a recipient of the message. An item in the list can either be a participant (exchange code and customer code) or an individual user (exchange code, customer code, and user signature). See user_code_t sub structure.
block_n	uint32_t Ignore. Always set to one.
bid_or_ask_c	uint8_t Indicates if the user wants a quote of the bid, ask or both sides of the market. Possible values: 0 = Both 1 = Bid 2 = Ask.
quote_status_c	uint8_t Indicates if quote request originated from a crossing quote request or a non-crossing quote request. Possible values: 'C' (decimal 67) = Crossing quote request 'V' (decimal 86) = Non-crossing quote request '0' (decimal zero) = Not defined.
filler_2_s	char[3] Ignore. Used for byte alignment.
mp_quantity_i	int64_t

Variable	Description
	Transaction quantity. A value of zero indicates that a quotation is requested without any volume.

28.3.2 user_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 participant. Participants can retrieve their own identifier using the omniapi_get_info_ex() function.

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®