

ASX Trade Refresh

ASX Trade Queries

June 2022





Table of Contents

1	Introduction			
	1.1	Software	Distribution Restrictions	22
	1.2	Supporte	d Platforms	22
	1.3	ASX Trad	e Support	
	1.4	ASX Trad	e OI Documentation Suite	
	1.5	Restrictio	ons	23
		1.5.1	Trading Participant Specific Information	23
		1.5.2	Broker Service Providers	23
	1.6	Version H	listory	24
2	Commo	n Structur	es	26
	21	Unique la	dentifiers - transaction type t and broadcast type t	26
	2.2	Series Str	ructure – series t	26
		001100 001		20
3	CQ14 R	ectified Ho	olding Trades Query	
	3.1	Query Fu	nction	28
	3.2	Query Pr	operties	28
	3.3	Answer P	Properties	28
	3.4	Message	Structure	28
		3.4.1	query_rectify_t	28
	3.5	Answer S	itructure	29
		3.5.1	answer_rectify_t_item_t	29
		3.5.2	ans_rect_t_item_t	29
		3.5.3	trading_code_t/user_code_t	31
		3.5.4	new_account_t/account_t	
4	CQ27 N	lissing Dea	ils Query	32
	4.1	Query Fu	nction	32
	4.2	Query Pr	operties	32
	4.3	Answer F	Properties	32
	4.4	Return C	odes	33
	4.5	Message	Structure	33
		4.5.1	cl_query_deal_var_t	33
	4.6	Answer S	itructure	34
		4.6.1	cl_answer_deal_var_hdr_t	34
		4.6.2	item_hdr_t	35
		4.6.3	sub_item_hdr_t	35
		4.6.4	cl_deal_base_t (named structure 24)	35
		4.6.5	time_spec_t	35
		4.6.6	cl_deal_asx_t (named structure 25)	
		4.6.7	cl_trade_public_asx_t (named structure 27)	36
		4.6.8	participant_t	37
		4.6.9	cl_trade_anonymous_asx_t (named structure 28)	37
		4.6.10	cl_deal_trade_report_asx_t (named structure 26)	
		4.6.11	cl_deal_trade_report_asx_item_t	



		4.6.12	cl_trade_cancel_asx_t (named structure 30)			
		4.6.13	cl_combo_series_asx_t (named structure 42)			
		4.6.14	cl_deal_extended_price_asx_t (named structure 34)			
		4.6.15	cl_deal_extended_price_asx_item_t			
5	CQ110	40				
	5 1	Query Fi	unction	40		
	5.2	Query P	ronerties	40		
	5.3	Answer	Properties			
	5.4	Message	e Structure			
		5.4.1	query dedicated trade t			
		5.4.2	party t			
	5.5	Answer	Structure	41		
		5.5.1	answer dedicated trade t	42		
		5.5.2	sub item hdr t	42		
		5.5.3	cl_trade_base_api_t (named structure 3)	42		
		5.5.4	trading_code_t/user_code_t	45		
		5.5.5	give_up_member_t	45		
		5.5.6	account_t	45		
		5.5.7	time_spec_t	46		
		5.5.8	asx_exchange_info_t	46		
		5.5.9	cl_trade_asx_api_t (named structure 65)	46		
		5.5.10	party_t	47		
		5.5.11	cl_deal_extended_price_asx_t (named structure 34)	47		
		5.5.12	cl_deal_extended_price_asx_item_tt	48		
		5.5.13	cl_deal_trade_report_asx_t (named structure 26)	48		
		5.5.14	cl_deal_trade_report_asx_item_t	48		
		5.5.15	cl_combo_series_asx_t (named structure 42)	48		
		5.5.16	cl_trade_regulatory_asx_t (named structure 35	49		
		5.5.17	ASX Specific Overlay of regulatory_data_s Variable	49		
		5.5.18	cl_trade_short_asx_t (named structure 36)			
6	CQ111 Dedicated Trade History Query					
	6.1	6.1 Ouery Function				
	6.2	Query P	roperties	50		
	6.3	Answer	Answer Properties			
	6.4	Message	e Structure	50		
		6.4.1	query_dedicated_trade_his_t	50		
		6.4.2	party_t	51		
	6.5	Answer	Structure	51		
		6.5.1	answer_dedicated_trade_his_t	52		
		6.5.2	sub_item_hdr_t	52		
		6.5.3	cl_trade_base_api_t (named structure 3)	52		
		6.5.4	trading_code_t/user_code_t	55		
		6.5.5	give_up_member_t	55		
		6.5.6	account_t	55		
		6.5.7	asx_exchange_info_t	56		
		6.5.8	cl_trade_asx_api_t (named structure 65)	56		
		6.5.9	party_t	57		



		6.5.10	cl deal extended price asx t (named structure 34)	57	
		6.5.11	cl deal extended price asx item t		
		6.5.12	cl_deal_trade_report_asx_t (named structure 26)		
		6.5.13	cl_deal_trade_report_asx_item_t		
		6.5.14	cl_combo_series_asx_t (named structure 42)		
		6.5.15	cl_trade_regulatory_asx_t (named structure 35)	59	
		6.5.16	ASX Specific Overlay of regulatory_data_s Variable	59	
		6.5.17	cl_trade_short_asx_t (named structure 36)	59	
7	CQ112	Clearing Ir	nstance	60	
	7.1	Query Fi	unction	60	
	7.2	Query P	roperties	60	
	7.3	Answer	Properties	60	
	7.4	Message	e Structure	60	
		7.4.1	query_cl_instance_t	60	
	7.5	Answer	Structure	61	
		7.5.1	query_cl_instance_t	61	
		7.5.2	answer_cl_instance_item_t	61	
8	CQ114	Initial Trac	de Reports Query	62	
	8.1	Query Fi	unction	62	
	8.2	Query P	roperties	62	
	8.3	8.3 Answer Properties			
	8.4	Message	e Structure	62	
		8.4.1	cl_query_itn_t	62	
	8.5	Answer	Structure	63	
		8.5.1	cl_answer_itn_var_hdr_t	63	
		8.5.2	item_hdr_t	63	
		8.5.3	sub_item_hdr_t	64	
		8.5.4	cl_deal_base_t (named structure 24)	64	
		8.5.5	time_spec_t	64	
		8.5.6	cl_deal_asx_t (named structure 25)	64	
		8.5.7	cl_trade_public_asx_t (named structure 27)	65	
		8.5.8	participant_t	66	
		8.5.9	cl_trade_internal_asx_t (named structure 29)	66	
		8.5.10	trading_code_t/user_code_t	67	
		8.5.11	account_t	67	
		8.5.12	give_up_member_t	68	
		8.5.13	<pre>cl_deal_trade_report_asx_t (named structure 26)</pre>	68	
		8.5.14	cl_deal_trade_report_asx_item_t	68	
		8.5.15	cl_trade_cancel_asx_t (named structure 30)	68	
		8.5.16	cl_deal_extended_price_asx_t (named structure 34)	69	
		8.5.17	cl_deal_extended_price_asx_item_t	69	
		8.5.18	cl_combo_series_asx_t (named structure 42)	69	
		8.5.19	cl_trade_regulatory_asx_t (named structure 35)	70	
		8.5.20	ASX Specific Overlay of regulatory_data_s Variable	70	
		8.5.21	cl_trade_short_asx_t (named structure 36)	70	



9	CQ1134	Dedicated	d Trade Proxy Query	71
	9.1	Query Fu	nction	71
	9.2	Query Pr	operties	71
	9.3	Answer P	Properties	71
	9.4	Message	Structure	71
	-	9.4.1	query dedicated trade t	
		942	narty t	72
	95	Answer S	party_t	72
	5.5	951	answer dedicated trade t	73
		9.5.1	sub item hdr t	73
		0 5 2	d trade base ani t (named structure 2)	
		9.5.5	trading code t/user code t	75
		9.J.4 0 E E	ridding_code_t/dsei_code_t	70
		9.5.5	give_up_inembel_t	70
		9.5.0		0 /
		9.5.7	time_spec_t	//
		9.5.8	asx_exchange_info_t	
		9.5.9	ci_trade_asx_api_t (named structure 65)	//
		9.5.10	party_t	
		9.5.11	cl_deal_extended_price_asx_t (named structure 34)	78
		9.5.12	cl_deal_extended_price_asx_item_t	79
		9.5.13	cl_deal_trade_report_asx_t (named structure 26)	79
		9.5.14	cl_deal_trade_report_asx_item_t	79
		9.5.15	cl_combo_series_asx_t (named structure 42)	79
		9.5.16	cl_trade_regulatory_asx_t (named structure 35)	80
		9.5.17	ASX Specific Overlay of Regulatory_data_s Variable	80
		9.5.18	cl_trade_short_asx_t (named structure 36)	80
10	CQ1135	Dedicated	d Trade History, Proxy Query	81
	10.1	Query Fu	nction	81
	10.2	Query Pr	operties	81
	10.3	Answer Properties		81
	10.4	Message	Structure	81
		10.4.1	query dedicated trade his t	81
		10.4.2	party t	82
	10.5	Answer S	tructure	82
		10.5.1	answer dedicated trade his t	83
		10.5.2	sub item hdr t	83
		10.5.3	cl trade base api t (named structure 3)	83
		10.5.4	trading code t/user code t	86
		10.5.5	give up member t	86
		10.5.6	account t	
		10 5 7	time spec t	87
		10.5.8	asx exchange info t	
		10.5 9	cl trade asx api t (named structure 65)	
		10 5 10	narty t	
		10 5 11	cl deal extended price asy t (named structure 3/1)	۵۵
		10.5.11	cl deal extended nrice ack item t	وo
		10.5.12	cl deal trade report asy t (named structure 26)	
		10 5 1/	cl deal trade report asy item t	وی ۵۰
		10.3.14		



		10.5.15	cl_combo_series_asx_t (named structure 42)	
		10.5.16	cl_trade_regulatory_asx_t (named structure 35)	90
		10.5.17	ASX Specific Overlay of regulatory_data_s Variable	90
		10.5.18	cl_trade_short_asx_t (named structure 36)	90
11	DQ6 Bro	oker Signat	ures Query	91
	11.1	Query Fur	nction	
	11.2	Query Pro	operties	
	11.3	Answer P	roperties	
	11.4	Message	Structure	
		11.4.1	query_broker_t	
	11.5	Answer St	tructure	92
		11.5.1	answer_broker_t	92
		11.5.2	answer_broker_item_t	92
		11.5.3	item struct for answer_broker_item_t	93
12	DQ7 Ma	arket		94
	12.1	Query Fur	nction	
	12.2	Query Pro	operties	
	12.3	Answer P	roperties	
	12.4	Message S	Structure	
		12.4.1	query_market_t	94
	12.5	Answer St	tructure	94
		12.5.1	answer_market_t	94
		12.5.2	answer_market_item_t	95
13	DQ8 Ins	trument G	roup	97
	13.1	Query Fur	nction	
	13.2	Query Pro	operties	97
	13.3	Answer P	roperties	97
	13.4	Message	Structure	97
		13.4.1	query_instrument_group_t	97
	13.5	Answer St	tructure	97
		13.5.1	answer_instrument_group_t	
		13.5.2	answer_instrument_group_item_t	98
14	DQ17 Ir	strument	Туре	
	14.1	Query Fur	nction	
	14.2	Query Pro	operties	
	14.3	Answer P	roperties	
	14.4	Message	Structure	
		14.4.1	query_instrument_t	
	14.5	Answer St	tructure	
		14.5.1	answer_ext_instrument_t	
		14.5.2	answer_ext_instrument_item_t	
15	DQ18 N	on Trading	Bays Query	
	15.1	Query Fur	nction	



	15.2	Query Properties	104
	15.3	Answer Properties	
	15.4	Message Structure	
		15.4.1 query_non_trading_days_t	
	15.5	Answer Structure	
		15.5.1 answer_non_trading_days_t	
		15.5.2 answer_non_trading_days_item_t	105
16	DQ24 I	Exchange	
	16.1	Query Function	
	16.2	Query Properties	
	16.3	Answer Properties	
	16.4	Message Structure	
		16.4.1 guery exchange dg24 t	
	16.5	Answer Structure	
		16.5.1 answer exchange da24 t	
		16.5.2 answer_exchange_da24_item_t	
17	DQ28 (Central Group	
	171	QueryEunction	108
	17.1	Query Pronerties	108
	17.2	Answer Properties	108
	17.5	Message Structure	108
	17.4	17.4.1 query central group t	108
	175	Answer Structure	108
	17.5	17.5.1 answer central group t	109
		17.5.2 answer_central_group_item_t	
18	DQ29 9	Session State Query	
	10 1	Query Function	110
	18.1	Query Pronerties	
	18.2	Answer Properties	110
	18.4	Message Structure	110
	10.4	18.4.1 query trading state t	
	18 5	Δnswer Structure	110
	10.5	18.5.1 answer trading state t	110
		18.5.2 answer_trading_state_item_t	
19	DQ30 (User Type Information Query	
	19.1	Query Function	
	19.2	Query Properties	
	19.3	Answer Properties	
	19.4	Message Structure	
		19.4.1 guery user type info t	
	19.5	Answer Structure	
		19.5.1 answer user type info t	
		19.5.2 answer user type info item t	



20	DQ33 Currency Query			
	20.1	Query Function		
	20.2	Query Properties	115	
	20.3	Answer Properties	115	
	20.4	Message Structure	115	
		20.4.1 guery currency t	115	
	20.5	Answer Structure		
		20.5.1 answer currency t		
		20.5.2 answer currency item t		
21	D045 Tr	rade Report Types Query	117	
	21.1		117	
	21.1	Query Properties		
	21.2	Answer Properties		
	21.5	Allswei Flöpeldes		
	21.4	21.4.1 guery trade report types t		
	21 5	21.4.1 query_trade_report_types_t		
	21.5	Answer Structure		
		21.5.1 answer_trade_report_types_t		
		21.5.2 answer_trade_report_types_item_t		
22	DQ46 D	eal Source Query	120	
	22.1	Query Function		
	22.2	Query Properties		
	22.3	Answer Properties		
	22.4	Message Structure		
		22.4.1 query deal source t	120	
	22 5	Answer Structure	120	
	22.5	22.5.1 answer deal source t	120	
		22.5.2 answer deal source_item t		
23	DQ53 Co	orporate Action Query		
	23.1	Query Function	122	
	23.2	Query Properties	122	
	23.3	Answer Properties	122	
	23.4	Message Structure	122	
		23.4.1 query_corp_action_t	122	
	23.5	Answer Structure	122	
		23.5.1 answer_corp_action_da53	123	
		23.5.2 answer_corp_action_da53_item_t	123	
24	DQ55 Cl	learing Participants Query	125	
	24.1	Query Function	125	
	24.2	Query Properties		
	24.3	Answer Properties		
	24.4	Message Structure		
		24.4.1 guery cl participant t		
	24.5	Answer Structure		
	-	24.5.1 answer_cl_participant_t		



		24.5.2	answer_cl_participant_item_t	126	
25	DQ56 Sp	ecial Liab	ility Query	127	
	25.1	Querv Fu	inction		
	25.2	Query Properties			
	25.3	Answer Properties			
	25.4	Message	Structure		
		25.4.1	query special liability t		
	25.5	Answer S	Structure		
		25.5.1	answer_special_liability_t		
		25.5.2	trading_code_t		
		25.5.3	answer_special_liability_item_t		
26	DQ57 M	ember Ob	oligation Query	129	
	26.1	Querv Fu	inction		
	26.2	Query Pr	operties		
	26.3	Answer P	Properties		
	26.4	Message	Śtructure		
		26.4.1	query member obligation t		
	26.5	Answer S	Structure		
		26.5.1	answer_member_obligation_da57_t		
		26.5.2	answer_member_obligation_da57_item_t	130	
27	DQ87 M	arket Mal	ker Protection Query	131	
	27.1	Query Fu	Inction		
	27.2	Query Pr	operties		
	27.3	Answer F	Properties		
	27.4	Message	Structure		
		27.4.1	query_mm_protection		
	27.5	Answer S	Structure	131	
		27.5.1	answer_mm_protection		
		27.5.2	answer_mm_protection_item_t		
28	DQ120 [Delta Und	erlying Query	133	
	28.1	Query Fu	inction	133	
	28.2	Query Pr	operties		
	28.3	Answer P	Properties		
	28.4	Related N	Messages		
	28.5	Message	Structure		
		28.5.1	query_delta_t	133	
		28.5.2	time_spec_t	134	
	28.6	Answer S	Structure	134	
		28.6.1	answer_segment_hdr_t	135	
		28.6.2	item_hdr_t	135	
		28.6.3	sub_item_hdr_t		
		28.6.4	ns_delta_header_t (named struct number 37001)		
		28.6.5	ns_remove_t (named structure 37002)		
		28.6.6	ns_underlying_basic_t (named structure 37201)		
		28.6.7	ns_tixed_income_t (named struct number 37202)		



		28.6.8	ns_coupon_dates_t – named struct number 37203				
29	DQ122	Delta Instr	rument Class Query				
	29.1	29.1 Query Function					
	29.2	Query Properties					
	29.3	Answer Properties					
	29.4	4 Related Messages					
	29.5	Message Structure					
		29.5.1	query_delta_t				
		29.5.2	time_spec_t				
	29.6	Answer S	Structure				
		29.6.1	answer_segment_hdr_t				
		29.6.2	item_hdr_t				
		29.6.3	sub_item_hdr_t				
		29.6.4	ns_delta_header_t (named structure 37001)				
		29.6.5	ns_remove_t (named structure 37002)				
		29.6.6	ns_inst_class_basic_t (named structure 37101)				
		29.6.7	ns_price_tick_t (named structure 37102)	147			
		29.6.8	tick_size_t				
		29.6.9	ns_block_size_t (named structure 37103)				
		29.6.10	ns_calc_rule_t (named structure 37104)				
		29.6.11	ns_crossing_rule_t (named structure 37106)	149			
		29.6.12	ns_inst_class_ext1_t (named structure 37107)	150			
30	DQ124	Delta Instr	rument Series				
	30.1	Query Fu	inction				
	30.2	Query Pr	operties				
	30.3	Answer H	Properties				
	30.4	Related I	Vlessages				
	30.5	Message	Structure				
		30.5.1	query_delta_t				
	20.0	30.5.2	time_spec_t				
	30.6	Answer S	structure				
		30.0.1	answer_segment_nur_t				
		30.0.Z	sub_item_hdr_t				
		30.0.3 20.6.4	sub_item_hui_t				
		2065	ns_remove_t (named structure 27002)				
		2066	ns_inct_sories_basis_t (named structure 37002)				
		30.0.0	ns_inst_series_basic_t (named structure 37301)				
		50.0.7	lis_list_selles_basic_sligle_t (liailied structure 57502)				
31	DQ126	Combinati	ion Series	159			
	31.1	Query Fu	Inction	159			
	31.2	Query Pr	operties	159			
	31.3	Answer F	Properties	159			
	31.4	Related N	Messages	159			
	31.5	Message	Structure	159			
		31.5.1	query_combo_t	159			
	31.6	Answer S	Structure				



		31.6.1	answer_segment_hdr_t	160
		31.6.2	item_hdr_t	
		31.6.3	sub_item_hdr_t	
		31.6.4	ns_inst_series_basic_t (named structure 37301)	
		31.6.5	ns_combo_series_leg_t (named structure 37308)	162
32	DQ127	Historical	Series	
	32.1	Ouerv Fu	unction	
	32.2	Query Pr	roperties	
	32.3	Answer I	Properties	
	32.4	Related I	Messages	
	32.5	Message	Structure	
		32.5.1	query_history_t	
		32.5.2	query_history_item_t	
	32.6	Answer S	Structure	
		32.6.1	answer_segment_hdr_t	164
		32.6.2	item_hdr_t	
		32.6.3	sub_item_hdr_t	
		32.6.4	ns_inst_series_basic_t (named structure 37301)	
		32.6.5	ns_inst_series_basic_single_t (named structure 37302)	166
33	DQ128	Historical	Combination Series	
	33.1	Ouerv Fi	inction	
	33.2	Ouerv Pr	roperties	
	33.3	Answer F	Properties	
	33.4	Related I	Messages	
	33.5	Message	Structure	
		33.5.1	query history t	
		33.5.2	query history item t	
	33.6	Answer S	Structure	
		33.6.1	answer segment hdr t	
		33.6.2	item hdr t	
		33.6.3	sub item hdr t	
		33.6.4	ns inst series basic t (named structure 37301)	
		33.6.5	ns_combo_series_leg_t (named structure 37308	172
34	ll17 Pre	eliminary S	ettlement Prices	174
	34.1	Ouerv Fu	unction	
	34.2	Ouerv Pr	roperties	
	34.3	Answer F	Properties	
	34.4	Message	structure	
		34.4.1	query prel settlement t	
	34.5	Answer S	Structure	
		34.5.1	answer prel settlement t	
		34.5.2	answer_prel_settlement_item_t	
35	IQ12 To	otal Equilib	prium Prices	176
	35 1		inction	176
	35.2	Ouerv Pr	roperties	176
		~~~~		



	35.3	Answer Properties		176
	35.4	Message	Structure	176
		35.4.1	query_tot_equil_prices_t	
	35.5	Answer S	Structure	
		35.5.1	answer_tot_equil_prices_t	
		35.5.2	answer_tot_equil_prices_item_t	177
36	IQ18 To	otal Volume	es and Query	179
	36.1	Query Fu	inction	179
	36.2	Query Pr	operties	
	36.3	Answer P	Properties	
	36.4	Message	Structure	
		36.4.1	query_hdr_t	
		36.4.2	sub_item_hdr_t	
		36.4.3	ob_levels_query_data_t (named structure 33020)	
		36.4.4	ob_levels_id_t (named structure 33002)	
	36.5	Answer S	Structure	
		36.5.1	answer_hdr_t	
		36.5.2	sub_item_hdr_t	
		36.5.3	ob_levels_next_query_t (named structure 33032)	
		36.5.4	ob_levels_id_t (named structure 33002)	
		36.5.5	ob_levels_price_volumes_t (named structure 33003)	
		36.5.6	ob_levels_price_volumes_item_t	
		36.5.7	ob_levels_order_number_t (named structure 33004)	
		36.5.8	ob_levels_total_quantity_t (named structure 33005)	
		36.5.9	ob_levels_no_of_orders_t (named structure 33033)	
		36.5.10	ob_levels_no_of_orders_item_t	
		36.5.11	market_info_base_t (named structure 33034)	
		36.5.12	market_info_asx_t (named structure 33039)	
		36.5.13	market_info_index_t (named structure 33040)	
		36.5.14	ob_levels_undisclosed_quantity_t (named structure 33041)	
		36.5.15	ob_levels_closing_t (named structure 33031)	
		36.5.16	ob_levels_price_t (named structure 33006)	
		36.5.17	ob_levels_price_item_t	
		36.5.18	market_info_asx_extended (named structure 33139)	
	36.6	Examples	S	
		36.6.1	Example 1	190
		36.6.2	Example 2	
		36.6.3	Example 3	
37	IQ19 To	otal Volume	es and Prices Query	
	37.1	Query Fu	inction	192
	37.2	Query Pr	operties	192
	37.3	Answer P	Properties	193
	37.4	Message	Structure	193
		37.4.1	query_hdr_t	193
		37.4.2	sub_item_hdr_t	193
		37.4.3	ob_levels_query_data_t (named structure 33020)	194
		37.4.4	ob_levels_id_t (named structure 33002)	194



	37.5	Answer S	Structure	
		37.5.1	answer_hdr_t	
		37.5.2	sub_item_hdr_t	
		37.5.3	ob_levels_next_query_t (named structure 33032)	
		37.5.4	ob_levels_id_t (named structure 33002)	
		37.5.5	ob_levels_price_volumes_t (named structure 33003)	
		37.5.6	ob_levels_price_volumes_item_t	
		37.5.7	ob_levels_order_number_t (named structure 33004)	
		37.5.8	ob_levels_total_quantity_t (named structure 33005)	
		37.5.9	ob_levels_price_t (named structure 33006)	
		37.5.10	ob_levels_price_item_t	
		37.5.11	ob_levels_no_of_orders_t (named structure 33033)	
		37.5.12	ob_levels_no_of_orders_item_t	
		37.5.13	market_info_base_t (named structure 33034)	
		37.5.14	market_info_asx_t (named structure 33039)	
		37.5.15	market_info_index_t (named structure 33040)	
		37.5.16	ob_levels_undisclosed_quantity_t (named structure 33041)	201
		37.5.17	ob_levels_closing_t (named structure 33031)	202
		37.5.18	market_info_asx_extended (named structure 33139)	202
	37.6	Examples	S	202
		37.6.1	Example 1	202
		37.6.2	Example 2	202
		37.6.3	Example 3	
38	IQ48 S	ettlement F	Prices Query	204
	38.1	Query Fu	inction	204
	38.2	Query Pr	operties	204
	38.3	Answer P	Properties	204
	38.4	Message	Structure	
		38.4.1	query_trade_statistics_settle_t	
		38.4.2	Return Codes	
	38.5	Answer S	Structure	205
		38.5.1	answer_trade_statistics_settle_t	
		38.5.2	answer_trade_statistics_settle_item_t	205
39	ML1 P	rice Limit Q	uery	206
	39.1	Querv Pr	operties	
	39.2	Answer F	Properties	
	39.3	Message	Structure	
		39.3.1	query limit trans t	
	39.4	Answer S	Structure	
		39.4.1	answer limit t	
		39.4.2	answer_limit_item_t	207
40	MQ7 T	otal Order	Book Query	208
	40.1		inction	200
	40.1		Snanchot Issue	200 کے۔۔۔۔۔ مرد
		40.1.1 40.1.2	Decommonded Practice	208
		40.1.2	No Donth Issue	208
		40.1.3	No Depth Issue	209



	40.2	Query Properties	
	40.3	Answer Properties	
	40.4	Message Structure	
		40.4.1 query_tot_ob_t	209
	40.5	Answer Structure	210
		40.5.1 answer_tot_ob_t	210
		40.5.2 answer_tot_ob_item_t	211
		40.5.3 order_no_id_t	211
		40.5.4 party_t	212
		40.5.5 Return Codes	
41	MQ8 T	otal Order Query	213
	41.1	Query Function	
	41.2	Query Properties	
	41.3	Answer Properties	
	41.4	Message Structure	
		41.4.1 query_tot_order_t	
		41.4.2 whose_t	
		41.4.3 trading_code_t	
	41.5	Answer Structure	
		41.5.1 answer_tot_order_t	215
		41.5.2 answer_tot_order_item_t	215
		41.5.3 party_t	216
		41.5.4 order_t	216
		41.5.5 order_var_t	217
		41.5.6 trading_code_t/ex_user_code_t	219
		41.5.7 give_up_member_t	220
		41.5.8 time_spec_t	
42	MQ9 T	otal Inactive Order Query	
	42.1	Query Function	
	42.2	Query Properties	
	42.3	Answer Properties	
	42.4	Message Structure	
		42.4.1 query_tot_order_t	
		42.4.2 whose_t	
		42.4.3 trading_code_t	
	42.5	Answer Structure	
		42.5.1 answer_tot_order_t	223
		42.5.2 answer_tot_order_item_t	223
		42.5.3 party_t	224
		42.5.4 order_t	224
		42.5.5 order_var_t	
		42.5.6 trading_code_t/ex_user_code_t	227
		42.5.7 give_up_member_t	228
		42.5.8 time_spec_t	
43	MQ78	Query Trade Reports	
	43.1	Query Function	229



	12.2		anartias	220
	43.2	Query Pr		
	43.3	Answer P	Properties	
	43.4		sucructure	
		43.4.1	query_tot_order_t	
		43.4.2	whose_t	
	40 F	43.4.3	trading_code_t	230
	43.5		answer trd report t	231
		43.5.1	answer_trd_report_t	
		43.5.2	answer_trd_report_item_t	
		43.5.3	trading_code_t	
		43.5.4	order_var_t	
		43.5.5	party_t	
		43.5.6	give_up_member_t	
		43.5.7	asx_exchange_info_t	
		43.5.8	ASX Specific Overlay of regulatory_data_s variable	235
44	MQ80	Unmatcheo	d Trade Reports Counterparty Query	236
	44.1	Query Fu	inction	236
	44.2	Query Pr	operties	236
	44.3	Answer P	Properties	236
	44.4	Message	Structure	236
		44.4.1	query_tot_party_t	236
	44.5	Answer S	Structure	237
		44.5.1	answer_trd_report_party_t	237
		44.5.2	answer_trd_report_item_t	237
		44.5.3	trading_code_t	238
		44.5.4	order_var_t	239
		44.5.5	party_t	240
		44.5.6	asx_exchange_info_t	240
		44.5.7	give_up_member_t	241
45	MQ92	Own Order	rs Query	242
	45.1	Query Fu	inction	242
	45.2	Query Pr	operties	242
	45.3	Answer F	Properties	242
	45.4	Message	Structure	242
		45.4.1	query order private filter t	242
		45.4.2	whose t	243
	45.5	Answer S	Structure	244
		45.5.1	answer order hdr t	244
		45.5.2	item hdr t	245
		45.5.3	sub item hdr t	245
		45.5.4	basic order book order t (named struct number 34817)	245
		45.5.5	trading code t	248
		45.5.6	give up member t	
		45.5.7	time spec t	248
		45.5.8	reserve order t (named structure 34812)	249
		45.5.9	centre point order t (named structure 34816)	249
		45.5.10	inactive order t (named structure 34818)	250
			` '	



		45.5.11	order_submitter_t (named structure 34819)	250
		45.5.12	ranking_time_t (named structure 34949)	250
		45.5.13	crossing_t (named structure 34820)	250
		45.5.14	regulatory_t (named structure 34821)	250
		45.5.15	short_sell_order_t (named structure 34829)	
		45.5.16	enhanced_cp_matching_t (named structure 34831)	251
46	MQ99	Maximum	Block Order Sizes Query	252
	46.1	Query Fu	unction	252
	46.2	Query Pr	operties	252
	46.3	Answer I	Properties	252
	46.4	Message	Structure	252
		46.4.1	query_block_size_t	252
	46.5	Answer S	Structure	252
		46.5.1	answer_block_size_t	252
47	MQ15	1 Query Or	der Broadcast	254
	47.1	Query Fu	unction	
	47.2	Query Pr	operties	
	47.3	Answer I	Properties	
	47.4	Message	structure	
		47.4.1	query order broadcast t	
		47.4.2	Return Codes	
	47.5	Answer S	Structure	
		47.5.1	answer hdr t	
		47.5.2	item_hdr_t	
		47.5.3	sub_item_hdr_t	
		47.5.4	query_order_broadcast_next_t (named structure 34911)	258
		47.5.5	block_price_trans_t (named structure 34007)	258
		47.5.6	block_price_trans_item_t	258
		47.5.7	hv_alter_trans_t (named structure 34010)	
		47.5.8	hv_alter_trans_p_t (named structure 34110)	261
		47.5.9	hv_order_trans_t (named structure 34005)	262
		47.5.10	hv_order_trans_p_t (named structure 34105)	263
		47.5.11	hv_price_2_trans_t (named structure 34001)	263
		47.5.12	multi_order_response_t (named structure 34906)	265
		47.5.13	order_change_combined_t (named structure 34902)	265
		47.5.14	order_change_separate_t (named structure 34903)	267
		47.5.15	order_chg_sep_trans_ack_t (named structure 34919)	269
		47.5.16	order_info_t (named structure 34917)	269
		47.5.17	order_t	270
		47.5.18	order_price_change_t (named structure 34905)	271
		47.5.19	order_return_info_t (named structure 34904)	272
		47.5.20	<pre>segment_instance_number_t (named structure 34901)</pre>	273
		47.5.21	order_trade_info_t (named structure 34920)	273
		47.5.22	order_leg_trade_info_t (named structure 34921)	274
		47.5.23	order_trade_info_asx_t (named structure 34922)	274
		47.5.24	trade_report_1_trans_t (named structure 34021)	275
		47.5.25	trade report 1 trans p t (named structure 34119)	



	47.5.26	trade_report_2_trans_t (named structure 34022)	277
	47.5.27	trd_rpt_cust_t	278
	47.5.28	exchange_info_t (named structure 50004)	278
	47.5.29	free_text_t (named structure 34801)	279
	47.5.30	clearing_info_t (named structure 34802)	279
	47.5.31	single_order_insert (named structure 34808)	279
	47.5.32	single_order_update_t (named structure 34809)	279
	47.5.33	basic_order_t (named structure 34810)	
	47.5.34	reserve_order_t (named structure 34812)	
	47.5.35	basic_order_update_t (named structure 34815)	
	47.5.36	centre_point_order_t (named structure 34816)	
	47.5.37	inactive_order_t (named structure 34818)	
	47.5.38	order_submitter_t (named structure 34819)	
	47.5.39	order_owner_t (named structure 34804)	
	47.5.40	ranking_time_t (named structure 34949)	
	47.5.41	crossing_t (named structure 34820)	
	47.5.42	regulatory_t (named structure 34821)	
	47.5.43	short_sell_order_t (named structure 34829)	
	47.5.44	short_sell_order_change_t (named structure 34830)	
	47.5.45	enhanced_cp_matching_t (named structure34831)	
	47.5.46	ex_user_code_t	
	47.5.47	give_up_member_t	
	47.5.48	order_var_t	
	47.5.49	party_t	
	47.5.50	trading_code_t	
	47.5.51	originator_trading_code_t	
	47.5.52	time_spec_t	
	47.5.53	asx exchange info t	
	47.5.54	ASX Specific Overlay of regulatory_data_s Variable	289
MQ154	Query Or	der Broadcast on Behalf	
40.4	0		204
48.1	Query Fu	Inction	
48.2	Query Pr	operties	
48.3	Answer F	Properties	
48.4	Message	Structure	
	48.4.1	query_order_broadcast_p_t	
	48.4.2	party_t	
	48.4.3	Return Codes	
48.5	Answer S	structure	
	48.5.1	answer_hdr_t	
	48.5.2	item_hdr_t	
	48.5.3	sub_item_hdr_t	
	48.5.4	query_order_broadcast_next_t (named structure 34911)	
	48.5.5	block_price_trans_t (named structure 34007)	
	48.5.6	block_price_trans_item_t	
	48.5.7	hv_alter_trans_t (named structure 34010)	297
	48.5.8	hv_alter_trans_p_t (named structure 34110)	298
	48.5.9	hv_order_trans_t (named structure 34005)	299
	48.5.10	hv_order_trans_p_t (named structure 34105)	
	48.5.11	hv_price_2_trans_t (named structure 34001)	

48



	48.5.12	multi_order_response_t (named structure 34906)	302
	48.5.13	order change combined t (named structure 34902)	302
	48.5.14	order change separate t (named structure 34903)	304
	48.5.15	order_chg_sep_trans_ack_t (named structure 34919)	306
	48.5.16	order_info_t (named structure 34917)	306
	48.5.17	order_t	307
	48.5.18	order_price_change_t (named structure 34905)	308
	48.5.19	order_return_info_t (named structure 34904)	309
	48.5.20	enhanced_cp_matching_t (named structure34831)	310
	48.5.21	segment_instance_number_t (named structure 34901)	310
	48.5.22	order_trade_info_t (named structure 34920)	310
	48.5.23	order_leg_trade_info_t (named structure 34921)	311
	48.5.24	order_trade_info_asx_t (named structure 34922)	311
	48.5.25	trade_report_1_trans_t (named structure 34021)	312
	48.5.26	trade_report_1_trans_p_t (named structure 34119)	313
	48.5.27	trade_report_2_trans_t (named structure 34022)	314
	48.5.28	trd_rpt_cust_t	315
	48.5.29	exchange_info_t (named structure 50004)	316
	48.5.30	free_text_t (named structure 34801)	316
	48.5.31	clearing_info_t (named structure 34802)	316
	48.5.32	single_order_insert_t (named structure 34808)	316
	48.5.33	single_order_update_t (named structure 34809)	317
	48.5.34	basic_order_t (named structure 34810)	317
	48.5.35	reserve_order_t (named structure 34812)	318
	48.5.36	basic_order_update_t (named structure 34815)	319
	48.5.37	centre_point_order_t (named structure 34816)	320
	48.5.38	inactive_order_t (named structure 34818)	321
	48.5.39	order_submitter_t (named structure 34819)	321
	48.5.40	order_owner_t (named structure 34804)	322
	48.5.41	ranking_time_t (named structure 34949)	322
	48.5.42	crossing_t (named structure 34820)	322
	48.5.43	regulatory_t (named structure 34821)	322
	48.5.44	short_sell_order_t (named structure 34829)	322
	48.5.45	short_sell_order_change_t (named structure 34830)	322
	48.5.46	ex_user_code_t	322
	48.5.47	give_up_member_t	323
	48.5.48	order_var_t	323
	48.5.49	party_t	325
	48.5.50	trading_code_t	325
	48.5.51	originator_trading_code_t	325
	48.5.52	time_spec_t	326
	48.5.53	asx_exchange_info_t	326
	48.5.54	ASX Specific Overlay of regulatory_data_s Variable	326
MQ392 ⁻	Total Orde	er for BSPs Query	328
49.1	Query Fu	nction	328
49.2	Query Pro	operties	328
49.3	Answer P	roperties	328
49.4	Message	Structure	328
	49.4.1	query_tot_order_p_t	328

49



		49.4.2	whose_t	
		49.4.3	trading_code_t	
	49.5	Answer S	tructure	
		49.5.1	answer_tot_order_t	
		49.5.2	answer_tot_order_item_t	
		49.5.3	party_t	
		49.5.4	order_t	
		49.5.5	order_var_t	
		49.5.6	trading_code_t/ex_user_code_t	
		49.5.7	give_up_member_t	
50	MQ393	Total Inac	tive Order for BSPs Query	
	50.1	Querv Fu	nction	
	50.2	Query Pro	operties	
	50.3	Answer P	, Properties	
	50.4	Message	Structure	
		50.4.1	query tot order p t	
		50.4.2	whose t	
		50.4.3	trading code t	
	50.5	Answer S	tructure	
		50.5.1	answer tot order t	
		50.5.2	answer tot order item t	
		50.5.3	party t	
		50.5.4	order t	
		50.5.5	order var t	
		50.5.6	trading code t/ex user code t	
		50.5.7	give up member t	
		50.5.8	time_spec_t	
51	MQ476	Own Orde	ers Query for BSPs	
	51 1	Ouerv Fu	nction	344
	51.2	Ouery Pro	operties	
	51.3	Answer P	Properties	
	51.4	Message	Structure	
		51.4.1	query order private filter p t	
		51.4.2	4	
	51.5	Answer S	tructure	
		51.5.1	answer order hdr t	
		51.5.2	item hdr t	
		51.5.3	sub item hdr t	
		51.5.4	basic order book order t (named struct number 34817)	
		51.5.5	trading code t	
		51.5.6	give_up_member_t	
		51.5.7	time spec t	
		51.5.8	reserve order t (named structure 34812)	
		51.5.9	enhanced cp matching t (named structure 34831)	
		51.5.10	centre point order t (named structure 34816)	
		51.5.11	inactive order t (named structure 34818)	
		51.5.12	order_submitter_t (named structure 34819)	



		51.5.13 ranking_time_t (named structure 34949)	352
		51.5.14 crossing_t (named structure 34820)	
		51.5.15 regulatory_t (named structure 34821)	
		51.5.16 ASX Specific Overlay of regulatory_data_s Variable	
		51.5.17 short_sell_order_t (named structure 34829)	353
52	UQ1 Pa	artition Query	
	52.1	Query Function	354
	52.2	Query Properties.	
	52.3	Answer Properties	
	52.4	Message Structure	354
	01.1	52.4.1 query partition t	
	52.5	Answer Structure	
		52.5.1 answer partition t	
		52.5.2 answer partition item t	355
		52.5.3 server_partition_t	
53	UO9 B	17 Signals Sent Ouerv	
	53.1	Query Function	
	53.2	Query Properties	356
	53.3	Answer Properties	
	53.4	Message Structure	
		53.4.1 query bi7 signals sent t	
	53.5	Answer Structure	
		53.5.1 answer bi7 signals sent t	
		53.5.2 answer_bi7_signals_sent_item_t	
54	UQ12 I	Business Date	
	54.1	Query Function	
	54.2	Query Properties.	358
	54.3	Answer Properties	
	54.4	Message Structure	358
	5	54.4.1 query husiness date t	358
	54.5	Answer Structure	
	0	54.5.1 answer_business_date_t	
55	UO14 F	BI81 Broadcasts Sent Ouerv	
	55 1	Query Function	360
	55.2	Query Properties	360
	55.3	Answer Properties	360
	55.4	Message Structure	360
	55.4	55.4.1 query hi81 broadcasts sent ta	360
	55 5	Answer Structure	
	55.5	55.5.1 answer segment hdr t	363
		55.5.2 item hdr t	267
		55.5.2 sub item hdr t	267
		$55.5.5 \pm 300_{10}$ [10] t message core infort (named structure 35001)	2CJ
		55.5.4 message_core_inito_t (named structure 35001)	2C3
		55.5.6 message information item t	



		55.5.7 destination_item_t (named structure 35003)	
		55.5.8 document_url_t (named structure 35004)	
56	UQ15 Ins	strument Status	
	56.1	Query Function	
	56.2	Query Properties	
	56.3	Answer Properties	
	56.4	Message Structure	
		56.4.1 query_instrument_status_t	
	56.5	Answer Structure	
		56.5.1 answer_instrument_status_t	
		56.5.2 answer_instrument_status_item_t	366
57	UQ20 BI	I73 Signals Sent	
	57.1	Query Function	
	57.2	Query Properties	
	57.3	Answer Properties	
	57.4	Message Structure	
		57.4.1 guery bi73 signals sent t	
	57.5	Answer Structure	
		57.5.1 answer bi73 signals sent t	
		57.5.2 answer_bi73_signals_sent_item_t	



### **1** Introduction

Queries are used to retrieve information from ASX Trade. Information that can be retrieved for example includes:

- Missing order and trade history
- The details of a participant's orders in the market
- Information about markets, instrument groups, types, classes and series
- Information about other reference data, such as currencies, trade report types and deal sources
- Information about prices and trade statistics
- The public order books for instrument series
- Trading and instrument session states
- Published market announcements.
- For each query, the following information is provided:
- Query Function information about the query
- Query Properties the facility required and the function call
- **Answer Properties** the structure name of the answer that ASX Trade will send in response to the query
- Message Structure the structure of the query including a list of variables, their type and description
- Answer Structure the structure of the answer including a list of variables, their type and description.

Queries provide a baseline snapshot of data which can then be kept up to date by applying any subsequent updates received in relevant broadcasts.

Some queries are restricted to certain user types. For example, Broker Service Providers (BSPs) can use certain queries to query restricted information on behalf of their clients. Those queries are not available to other user types.

#### 1.1 Software Distribution Restrictions

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

#### **1.2 Supported Platforms**

The following platforms are supported by ASX Trade:

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

### **1.3** ASX Trade Support

For ASX Trade Open Interface Support, contact ASX Customer Technical Support (CTS) team either via email on <a href="https://cts.org/actional.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/licenses/by-non-cts.org/lice

#### 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 trading participant's request, 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 Providers (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	August 2014	<ul> <li>XT added in the condition code mapping table for item 79 in ASX Signals/Reference Point column.</li> </ul>
v1.2	August 2014	<ul> <li>Item 39.5.5 Return Codes, second line in table changed to: Cstatus - Transaction aborted Txstatus - ME_OHS_DATAINCOMPLETE- Snapshot could not be returned</li> </ul>
v2.0	March 2015	<ul> <li>CQ110 - Changes made to cl_trade_asx_api_t (named structure 65)</li> <li>CQ111 - Changes made to cl_trade_asx_api_t (named structure 65)</li> <li>CQ114 - Changes made to cl_trade_internal_asx_t (named structure 29)</li> <li>CQ1134 - Changes made to cl_trade_asx_api_t (named structure 65)</li> <li>CQ1135 - Changes made to cl_trade_asx_api_t (named structure 65)</li> <li>CQ1135 - Changes made to cl_trade_asx_api_t (named structure 65)</li> <li>MQ92 - New sub structure added - enhanced_cp_matching_t (named structure 34831) and changes made to centre_point_order_t (named structure 34816).</li> <li>MQ476 - New sub structure added - enhanced_cp_matching_t (named structure 34816)</li> <li>MQ151 - Changes made to order_trade_info_asx_t (named structure 34922) and centre_point_order_t (named structure 34831)</li> <li>MQ151 - New sub structure added - enhanced_cp_matching_t (named structure 34831)</li> <li>MQ151 - New sub structure added - enhanced_cp_matching_t (named structure 34831)</li> <li>MQ154 - Changes made to order_trade_info_asx_t (named structure 34922) and centre_point_order_t (named structure 34816).</li> <li>MQ154 - New sub structure added - enhanced_cp_matching_t (named structure 34831)</li> <li>MQ154 - New sub structure added - enhanced_cp_matching_t (named structure 34831)</li> <li>Trade condition code mapping table has been updated with new deal source values.</li> <li>New trade types have been added to the Mapping Table.</li> </ul>
v2.1	October 2018	<ul> <li>Updated to new ASX branding</li> <li>Removal of market, instrument group and trade condition code appendices, which are now covered in <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u>.</li> <li>Correction of the definition of passive and aggressive in aggressive_c in order_trade_info_asx_t</li> <li>Removal of references to ASX BookBuild</li> </ul>
v2.2	July 2019	<ul> <li>Removal of duplicate named struct entries on p253 and p289</li> <li>Corrected centre_point_order_t named struct ID on p253 and p290.</li> </ul>
v3.0	September 2019	<ul> <li>Updated for ASX Trade Refresh         <ul> <li>References to activation of inactive orders removed</li> <li>UQ14 Business Date logic updated</li> <li>Instance_c logic updated</li> </ul> </li> </ul>



Version	Date	Comment
		<ul> <li>Updated variable combo_mark_c description</li> <li>Updated variable asof_time_s description</li> <li>Updated variable created_time_s description</li> </ul>
v3.1	October 2019	<ul> <li>Updated variable order_filter_i description</li> <li>Updated instance_c logic for CQ110 and CQ111</li> <li>Updated instance_next_c logic for CA110 and CA111</li> </ul>
v3.2	November 2019	<ul> <li>Updated for ASX Trade Refresh</li> <li>Operating System Support for Rhel6.10 x86 (32 and 64 bit)</li> </ul>
v3.3	January 2020	<ul> <li>Removal of references to Deal Capture subsystem</li> <li>Change in IQ18/19 logic description in section 35.1</li> <li>Addition of market_info_asx_extended (named structure 33139) under IQ18/19 broadcasts</li> </ul>
v3.4	February 2020	<ul> <li>Updated instance_next_c description for CQ114, CQ1134 &amp; CQ1135</li> <li>Updated instance_c description for CQ114, CQ1134 &amp; CQ1135</li> </ul>
v3.5	April 2020	Addition of DQ87 Market Maker Protection Query in section 27
v3.6	August 2020	Update to MP_OHS_DATAINCOMPLETE
v3.7	October 2020	<ul> <li>Sequence number description updated for the following sections:</li> <li>4.6.4, 5.5.3, 6.5.3, 8.5.4, 9.5.3 and 10.5.3</li> </ul>
v3.8	December 2020	40.1.1 Snapshot issue updated
v3.9	April 2021	<ul> <li>Updated ex_customer_s variable description in section 9.4.2</li> <li>Updated section 47.2 Query Properties description</li> </ul>
v4.0	August 2021	<ul> <li>Updated inc_id_s data type in section 29.6.6</li> <li>Updated note for section 4.1 Query Function</li> <li>Updated description in section 33.1</li> <li>Updated description of variable date_trading_s in section 33.5</li> </ul>
v4.1	June 2022	Updated series_t description in section 47.4.1



### 2 Common Structures

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

### 2.1 Unique Identifiers - transaction_type_t and broadcast_type_t

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

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

#### 2.2 Series Structure – series_t

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

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



### 3 CQ14 Rectified Holding Trades Query

### 3.1 Query Function

This query is used to obtain all the rectified (cancelled) trades that are either currently in a holding state as the counterparty has yet to cancel their side of the trade, or that have been cancelled.

### 3.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP7	
Struct Name	query_rectify_t	
Partitioned	true	
Segmented	true	
Answers	CA14	

#### 3.3 Answer Properties

Transaction Type	CA14
Struct Name	answer_rectify_t

### 3.4 Message Structure

### 3.4.1 query_rectify_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 14}.
series	series_t Zero fill all fields.
instance_c	uint8_t Instance number - in the first query set to 1, and for subsequent queries, the instance_next_c returned in the answer is to be used.
filler_1_s	char[1] Ignore. Used for byte alignment.
segment_number_n	uint16_t This field indicates which segment users wish to receive, typically starting at 1. The reply structure has a segment number field too, which indicates which segment was returned.
search_series	search_series_t Same struct definition as series_t. Fill in country number, market code and instrument group to retrieve information on that particular instrument type. Zero fill to search for all rectified holding trades.



### 3.5 Answer Structure

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 14}.
segment_number_n	uint16_t Indicates which segment was returned. Contains zero if this is the last segment.
reserved_2_s	char[2] Ignore.
partition_low	partition_low_t Same struct definition as series_t. Indicates the series binary code that is the lowest range for this partition.
partition_high	partition_high_t Same struct definition as series_t. Indicates the series binary code that is the highest range for this partition.
items_n	uint16_t Indicates the number of items in the array.
instance_next_c	uint8_t Ignore. Not used.
filler_1_s	char[1] Ignore. Used for byte alignment.
item	answer_rectify_t_item_t[400] See answer_rectify_t_item_t sub structure below.

### 3.5.1 answer_rectify_t_item_t

Variable	Description
ans_rect_t_item	ans_rect_t_item_t Users should not question as to why this is so.
	See ans_rect_t_item_t sub structure below.

### 3.5.2 ans_rect_t_item_t

Variable	Description
created_date_s	char[8]
	The date when the user cancelled this side of the trade.
	Format: YYYYMMDD.
created_time_s	char[6]
	The time when the user cancelled this side of the trade.
	Format: HHMMSS.
asof_date_s	char[8]
	The date when the trade was created.
	Format: YYYYMMDD.
asof_time_s	char[6]
	The time of matching.
	Format: HHMMSS.



Variable	Description
clearing_date_s	char[8] The date of clearing in YYYYMMDD format.
orig_clearing_date_s	char[8] The original date of clearing in YYYYMMDD format.
trading_code	trading_code_t User who rectified (cancelled) the trade. See trading_code_t sub structure.
user_code	user_code_t User who rectified (cancelled) the trade. See user_code_t sub structure.
series	series_t The series of the trade.
trade_number_i	int32_t Ignore. Currently not used.
rectify_trade_number_i	int32_t Indicates the sequence number of rectified holding trades for this user.
ext_seq_nbr_i	uint32_t ASX Trade Slip number. Of the format 10PNNNNNN: 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC NNNNNN - Trade number.
state_c	uint8_t The state of the trade rectification. 1 = Holding. The counterparty to the trade has not yet cancelled their side. 5 = Completed. The counterparty to the trade has cancelled their side.
bought_or_sold_c	uint8_t The side of the trade that is cancelled. Possible values include: 1 = Bid 2 = Ask.
reserved_prop_c	uint8_t Ignore. Not used by ASX Trade.
filler_1_s	char[1] Ignore. Used for byte alignment.
new_account	new_account_t See new_account_t sub structure below.
account	account_t See account_t sub structure.
trade_quantity_i	int64_t Quantity of the trade.
deal_price_i	int32_t The price at which the trade occurred.
trade_short_sell_quantity_i	int64_t



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

### 3.5.3 trading_code_t/user_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this field 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 three 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 cancelled the trade. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

### 3.5.4 new_account_t/account_t

Variable	Description
country_id_s	char[2]
	For ASX Trade this field 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 three 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.
account_id_s	char[10]
	The account identifier.
filler_3_s	char[3]
	Ignore. Used for byte alignment.



### 4 CQ27 Missing Deals Query

#### 4.1 Query Function

This query is used to recover missing trade information. It can also be used to retrieve trades that occurred while the participant was not signed on, or during the participant's recovery process. The answer to a CQ27 query returns data in the format of a CB16 broadcast.

Participants should be aware that the CB16 broadcasts they receive depend on their participant type configured in ASX Trade. For instance, a participant who is only allowed to access equities and warrants will only receive CB16 broadcasts relating to those types of instruments. Therefore, they will be noticing gaps in the broadcasts sequence numbers and should not infer from these gaps that they have missed broadcasts and hence require beginning their recover process.

Furthermore, the CQ27 query relies on the cache of the CB16 broadcasts that are stored at the gateway.

For instance, if participant A only has access to equities and warrants, and participant B only has access to derivatives, then the CB16 broadcasts they receive would reflect their system configured rights. If both participants were to execute a CQ27 query to retrieve all trades for that day, then the gateway would respond with all the trade broadcasts stored in that one buffer, effectively providing both users with trades that do not concern them. The Open Interface (OI) application must handle this situation. Filtering the query by instrument type would be one way to do this.

The above scenario becomes slightly more complex for derivative only traders recovering trades that involve Tailor Made Combinations (TMCs) containing a mixture of equity and derivative legs. When there are no other types of participants on the same gateway, then the gateway cache will contain the derivative trade broadcasts plus any equity trade broadcasts that relate to the equity legs of a TMC. However, if they were to access a gateway cache that has all the equity trades in it due to the presence of another type of participant, then the derivative type participants need to determine the difference between normal equity trades and ones that result from a TMC. To do this, they need to parse each retrieved CB16 broadcast and look for the cl_combo_series_asx_t struct. This struct indicates that it originated from the equity leg of a TMC.



### Note:

When querying for today's trades, use facility type EP7. When querying for trades from T-1, you may use facility type EP5.

#### 4.2 Query Properties

Function Call	omniapi_query_ex
Facility	EP7 for today's trades and EP5 for T-1 trades
Struct Name	cl_query_deal_var_t
Partitioned	False
Segmented	False
Answers	CA27

#### 4.3 Answer Properties

|--|--|



C.L.		-	- N I	-		-
- NTI	r I I	CT	- 13	-	rn	$\boldsymbol{\omega}$
		~ ~				<u> </u>

The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs are described below.

#### 4.4 Return Codes

A CQ27 query may also be aborted by the DC subsystem, in which case only the reason for the transaction being aborted is returned to the sender.

Cstatus	Txstat
Successful	0 – All items returned.
Transaction aborted	CL_OMN_DATAPURGED – Trade history cache is purged and is not available.
Transaction aborted	CL_OMN_DATAINCOMPLETE – Trade history cache is incomplete or inconsistent with respect to this query. The gateway is currently recovering data.
Transaction aborted	CL_OMN_DATAINCOMPLETE_NORECOV – Trade history cache is incomplete and the gateway cannot recover it from the central system.



Note:

CB16 broadcasts are recovered by this query.

### 4.5 Message Structure

#### 4.5.1 cl_query_deal_var_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 27}.
series	series_t Zeros act as a wildcard to retrieve information from all markets. Fill in country number and market code to retrieve information on that particular market. Fill in a complete instrument type, instrument class or instrument series to retrieve information at that respective level. Fill only the commodity field to retrieve information on that underlying.
instance_c	uint8_t This is the instance number of the Trade Handler process. As Trade Handler is not partitioned this value should be set to one. Sequence numbers are per day. A user should query for all trades using sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero (0) and the last trade is sent in the answer. If there are no more trades to query instance_next_c in the answer is set to zero indicating that there is no more data to query.
filler_3_s	char[3] Ignore. Used for byte alignment.
sequence_first_u	uint32_t Defines the first deal of the range within which trades are sought. The sequence numbers start from one every day. If the maximum number of items for one transaction is returned, the query should be repeated with the next missing sequence number as first argument.
sequence last u	uint32 t



Variable	Description
	Defines the last trade of the range within which trades are sought. The sequence numbers starts from one every day. A value of zero indicates that the user wishes to retrieve all available trades. If the number of items exceeds the maximum that can fit into one answer, the query should be repeated with the next missing sequence number as first argument.
date_s	char[8] This must be the current or previous trading date. If querying for today's trades, use facility EP7. If querying for trades from T-1, use facility EP5. Format: YYYYMMDD.

### 4.6 Answer Structure

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

- cl_answer_deal_var_hdr_t
  - one or more sequences of:
  - item_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).

The answer is built like a CB16 broadcast but with several trades returned. Each item in the answer represents one trade. The only time that this is not the case is when a trade from an equity/derivative combination is retrieved. The trade information of the equity leg is appended to the same query item returning the first derivative leg.

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 27}.
items_n	uint16_t The number of top level items held in this structure.
instance_next_c	uint8_t Ignore. Not used.
filler_1_s	char[1] Ignore. Used for byte alignment.

### 4.6.1 cl_answer_deal_var_hdr_t



### 4.6.2 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.

### 4.6.3 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.

### 4.6.4 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 Ignore. Not used.
filler_1_s	char[1] Ignore. Used for byte alignment.

### 4.6.5 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.



### 4.6.6 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
	NNNNNN: Trade number.
exch_order_type_n	uint16_t
	Exchange specific order types.
	8 = Price Stabilisation/Green Shoe order
	64 = Centre Point order
	Refer to ASX Trade Markets, Instrument Groups and Trade Condition Codes.
trade_condition_n	uint16_t
	The condition in which a trade was executed. Possible values include:
	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.
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.
	For other values refer to <u>ASX Trade Introduction and Business Information</u> – Trade Report
	Types.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

### 4.6.7 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. Always set to one.
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 include: 1 = Bought 2 = Sold.
deal_source_c	int16_t


Variable	Description
	Where the trade was created. <u>Refer to ASX Trade Introduction and Business Information</u> – Trade Source.
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 Further information on the trade. Possible values include: 0 = Ignore 32 = 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.

# 4.6.8 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.

## 4.6.9 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. Always set to 1.
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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the trade was created. Refer to <u>ASX Trade Introduction and Business Information</u> – Trade Source for the possible values in this field.
filler_2_s	char[2] Ignore. Used for byte alignment.
trade_number_i	int32_t



Variable	Description
	A number that can be used to relate information in this broadcast with the associated information in a CB15.
big_attention_u	uint32_t Further information on the trade. Possible values include: 0 = Ignore 32 = 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.

## 4.6.10 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.

# 4.6.11 cl_deal_trade_report_asx_item_t

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

## 4.6.12 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.



Variable	Description
trade_type_c	uint8_t Always set to four, indicating a reversed trade.
filler_3_s	char[3] Ignore. Used for byte alignment.

#### 4.6.13 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB16 when the trade or trade report 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.

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

#### 4.6.14 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] 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.

#### 4.6.15 cl_deal_extended_price_asx_item_t

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



## 5 CQ110 Dedicated Trade Query

### 5.1 Query Function

The dedicated trade queries CQ110 and CQ111 can be used in conjunction with CB15 to recover a participant's own trades. When a missing sequence number is detected, these queries are used for the purpose of synchronisation. CQ110 returns data in the format of the CB15 broadcast.

## 5.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP7	
Struct Name	query_dedicated_trade_t	
Partitioned	False	
Segmented	False	
Answers	CA110	

#### 5.3 Answer Properties

Transaction Type	CA110
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead the sequence of possible structs is described below.

#### 5.4 Message Structure

## 5.4.1 query_dedicated_trade_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 110}.
series	series_t Not used in this query. Set to zeros.
instance_c	uint8_t This is the instance number of the Trade Handler process. As Trade Handler is not partitioned this value should be set to one. Sequence numbers are per day. A user should query for all trades using sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero (0) and the last trade is sent in the answer. If there are no more trades to query instance_next_c in the answer is set to zero indicating that there is no more data to query.
filler_3_s	char[3] Ignore. Used for byte alignment.
sequence_first_i	int32_t Defines the first trade of the range within which trades are sought. The sequence numbers starts from one every day. If the number of items exceeds the maximum that fits into one



Variable	Description
	answer, the query should be repeated with the next missing sequence number as first argument.
sequence_last_i	int32_t Defines the last trade of the range within which trades are sought. The sequence numbers starts from one every day. A value of zero indicates that the user wishes to retrieve all available trades. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as first argument.
date_s	char[8] This must be the current date. Format: YYYYMMDD.
party	party_t See party_t substructure below. Must be filled with blanks in this query.

## 5.4.2 party_t

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

#### 5.5 Answer Structure

The answer is built like a CB15 broadcast but with several trades returned. Each item in the answer represents one trade. The only time that this is not the case is when trades from an equity/derivative combination are retrieved. The trade information of the equity leg will be appended to the same CB15 broadcast as the first derivative leg. A trade consists at least the structure cl_trade_base_api_t.

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

- answer_dedicated_trade_t
  - one or more sequences of:
  - item_hdr_t
    - 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).

### 5.5.1 answer_dedicated_trade_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 110}.
instance_next_c	uint8_t Set to zero or one. Used to denote the next instance_c to query. If set to zero there is no more data to query.
filler_1_s	char[1] Ignore. Used for byte alignment.
items_n	uint16_t The number of top level items held in this structure.

# 5.5.2 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.

## 5.5.3 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.



Variable	Description
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 participants 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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the trade is created. Refer to <i>ASX Trade Introduction and Business Information</i> – <i>Trade Source</i> for the possible values in this field.
open_close_req_c	uint8_t Ignore. Set to zero.
trade_type_c	uint8_t Defines the type of trade. Possible values: 1 = Standard - The trade is a normally 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 customer (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. 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).



Variable	Description
attention c	uint8 t
dttention_e	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.
exchange_info_s	char[32]
	A free text field used at participant's discretion.
	For trade reports and Centre Point trades, 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
	Further information on the trade. Possible values include:
	U = Ignore
	32 = 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
alaaring data c	
clearing_date_s	Charles The date of clearing in XXXXMMDD format
execution_timestamp	time_spec_t Time the trade was entered into the system or execution time on market trades. See
	time specific table was entered into the system of execution time of market trades. See
trade venue c	
trade_venue_c	Defines the trade venue i.e. from where the trade emanates
instance c	
listance_c	Ignore. Not used.
exch order type n	
excil_order_type_fr	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 = IVIarKet BIG order
	(premium_r = un integer, order_type_c = 1, only entered by ASX trading Operations). 8 = Price Stabilisation/Green Shoe order
	(nremium i = an integer order type c = 1 time validity n l= 0)
	32 = Undisclosed (use order type c to determine order type)
	64 = Centre Point order (use order_type_c to determine market or limit)



Variable	Description
	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 Ignore. Not used by ASX Trade.
filler_2_s	char[2] Ignore. Used for byte alignment.
match_id	match_id_t Ignore. Not used by ASX Trade.

## 5.5.4 trading_code_t/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 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.

# 5.5.5 give_up_member_t

country_id_s

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 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.
5.5.6 account_t	
Variable	Description

For ASX Trade this is always set to "AU", indicating the Australian exchange.

char[2]



Variable	Description
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 3 digit number whereas information vendors have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
account_id_s	char[10] The account identifier.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 5.5.7 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.	

# 5.5.8 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.

# 5.5.9 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.



Variable	Description
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 1OPNNNNNN: 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC
trade_condition_n	<pre>uint16_t The condition in which a trade was executed. Possible values include: 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</pre>
	values. Refer to <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u> .
ext_t_state_c	uint8_t Trade report codes. Possible value includes: 0 = Ignore, not relevant. For other values refer to ASX Trade Introduction and Business Information - Trade Report Types for more information.
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.

### 5.5.10 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 3 digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
filler_1_s	char[1] Ignore. Used for byte alignment.

# 5.5.11 cl_deal_extended_price_asx_t (named structure 34)

This is not a mistake; this is the same structure as used in cl_deal_trade_report_asx_t. 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



Variable	Description
	This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the undefined value indicating that it is to be ignored. 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_extended_price_asx_item_t[3] There are always three items in the array. See cl_deal_extended_price_asx_item_t substructure below.

### 5.5.12 cl_deal_extended_price_asx_item_t

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

## 5.5.13 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.

## 5.5.14 cl_deal_trade_report_asx_item_t

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

## 5.5.15 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB15 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.



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

## 5.5.16 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.

### 5.5.17 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.

## 5.5.18 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.



# 6 CQ111 Dedicated Trade History Query

## 6.1 Query Function

This query works similar to CQ110 and is used to recover trades from T-1.

## 6.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP5	
Struct Name	query_dedicated_trade_his_t	
Partitioned	False	
Segmented	False	
Answers	CA111	

## 6.3 Answer Properties

Transaction Type	CA111
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the
	sequence of possible structs is described below.

### 6.4 Message Structure

## 6.4.1 query_dedicated_trade_his_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 111}.
series	series_t Not used in this query. Set to zeros.
instance_c	uint8_t This is the instance number of the Trade Handler process. As Trade Handler is not partitioned this value should be set to one. Sequence numbers are per day. A user should query for all trades using sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero (0) and the last trade is sent in the answer. If there are no more trades to query instance_next_c in the answer is set to zero indicating that there is no more data to query.
filler_3_s	char[3] Ignore. Used for byte alignment.
from_date_s	char[8] From date. Must be set to the previous trading date. Format: YYYYMMDD.
sequence_first_i	int32_t



Variable	Description
	Defines the first trade of the range within which trades are sought. The sequence numbers start from one every day. If the maximum number of items for one transaction is returned, the query should be repeated with the next missing sequence number as first argument.
to_date_s	char[8] To date. Format: YYYYMMDD.
sequence_last_i	int32_t Defines the last trade of the range within which trades are sought. The sequence numbers start from one every day. A value of zero indicates that the user wishes to retrieve all available trades. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as first argument.
party	party_t See party_t substructure below. Must be filled with blanks in this query.

## 6.4.2 party_t

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

#### 6.5 Answer Structure

The answer is built like a CB15 broadcast but with several trades returned. Each item in the answer represents one trade. The only time that this is not the case is when trades from an equity/derivative combination are retrieved. The trade information of the equity leg will be appended to the same CB15 broadcast as the first derivative leg. A trade consists of at least the structure cl_trade_base_api_t.

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

- answer_dedicated_trade_his_t
  - one or more sequences of:
  - item_hdr_t:
    - 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).

## 6.5.1 answer_dedicated_trade_his_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 111}.
from_date_s	char[8] From date. Format: YYYYMMDD.
sequence_first_i	int32_t Defines the first trade of the range within which trades are sought. The sequence numbers start from one every day. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as the first argument.
instance_next_c	uint8_t Set to zero or one. Used to denote the next instance_c to query. If set to zero there is no more data to query.
filler_1_s	char[1] Ignore. Used for byte alignment.
items_n	uint16_t The number of top level items held in this structure.

## 6.5.2 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.

## 6.5.3 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.



Variable	Description
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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the trade is created. Refer to <i>ASX Trade Introduction and Business Information -</i> <i>Trade Report Types</i> for the possible values in this field.
open_close_req_c	uint8_t Ignore. Set to zero.
trade_type_c	uint8_t Defines the type of trade. Possible values include: 1 = Standard. The trade is a normally 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 customer (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. 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 will be the same as created_date_s.
asof_time_s	char[6] The time of matching.
modified_date_s	char[8]



Variable	Description
	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.
exchange_info_s	char[32] A free text field used at the participant's discretion. For trade reports and Centre Point trades, 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 Further information on the trade. Possible values include: 0 = Ignore 32 = 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.
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.
instance_c	uint8_t Ignore. Not used.
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. Can only be a sell order  (bid_or_ask_c = 2). 4 = Market Bid order (only entered by ASX Trading Operations) 8 = Price Stabilisation/Green Shoe order



Variable	Description
	64 = Centre Point order
	2048 = Sweep order
	4096 = Centre Point Block order.
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
	Ignore. Not used by ASX Trade.
filler_2_s	char[2]
	Ignore. Used for byte alignment.
match_id	match_id_t
	Ignore. Not used by ASX Trade.

# 6.5.4 trading_code_t/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 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.

# 6.5.5 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 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.

# 6.5.6 account_t

Variable	Description
country_id_s	char[2]



Variable	Description
	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.

## 6.5.7 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.

# 6.5.8 cl_trade_asx_api_t (named structure 65)

Variable	Description
party	party_t This is the declared counter party 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 O - last digit of the Ordinal date



Variable	Description
	P - Instance number of DC NNNNNN: Trade number.
trade_condition_n	uint16_t The condition in which a trade was executed. Possible values include: 0 = No Condition 2 = Internal Trade/Crossing 8 = Buy Write (Equity/Derivative Combination). This field acts as a bit mask. The binary <b>AND</b> operator can be used on the above possible values. Refer to <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u> .
ext_t_state_c	uint8_t Trade report codes. Possible values include: 0 = Ignore, not relevant. For other values refer to ASX Trade Introduction and Business Information - Trade Report Types for more information.
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.

# 6.5.9 party_t

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

### 6.5.10 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 undefined 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_extended_price_asx_item_t[3]



Variable	Description
	There are always three items in the array.
	See cl_deal_extended_price_asx_item_t substructure below.

## 6.5.11 cl_deal_extended_price_asx_item_t

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

## 6.5.12 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.

## 6.5.13 cl_deal_trade_report_asx_item_t

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

### 6.5.14 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB15 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.

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



## 6.5.15 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.

### 6.5.16 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_da	a_s character position	s are to be padded by	y 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 (default).
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.

## 6.5.17 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.



### 7 CQ112 Clearing Instance

#### 7.1 Query Function

The Trade Handler subsystem is not partitioned. The UQ1 query can be used to retrieve partition information for the Matching Engine subsystem. However the OI does **not** contain a query that retrieves similar information for the TH subsystem.

What can be retrieved using the CQ112 query is each individual series and the associated TH partition from which trades for that series will be broadcast. Filtering the query on series (and only series - **not** class, underlying, type, group or market) will return to the OI user information for that specific series. Filtering the query on instance will return all the series that will have trades broadcast from that partition.

The TH subsystem in ASX Trade is not partitioned in the same way as the Matching Engine subsystem, meaning that the number of TH partitions and their underlying range allocation are not the same. Therefore, if a series is in the first partition of the Matching Engine it will notbe in the first partition of TH.

Function Call	omniapi_query_ex	
Facility	EP7	
Struct Name	query_cl_instance_t	
Partitioned	true	
Segmented	true	
Answers	CA112	

#### 7.2 Query Properties

#### 7.3 Answer Properties

The answer is a list of series with corresponding Trade Handler instance numbers.

Transaction Type	CA112
Struct Name	answer_cl_instance_t

#### 7.4 Message Structure

#### 7.4.1 query_cl_instance_t

Variable	Description
transaction_type	transaction_type_t
	Set the structure to the following: { C , Q , 112}.
series	series_t
	Zeros act as a wildcard to retrieve all series belonging to a certain DC partition. Fill in the complete instrument series to retrieve information at that respective level.
instance_c	uint8_t
	Set to zero
filler_1_s	char[1]
	Ignore. Used for byte alignment.



Variable	Description
segment_number_n	uint16_t This field indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.

## 7.5 Answer Structure

## 7.5.1 query_cl_instance_t

Variable	Description
transaction_type	transaction_type_t
	Contains the following: {'C', 'A', 112}.
instance_next_c	uint8_t
	Ignore. Not used.
filler_3_s	char[3]
	Ignore. Used for byte alignment.
segment_number_n	uint16_t
	Indicates which segment was returned. Contains zero if this is the last segment.
item	answer_cl_instance_item_t[1000]
	See answer_cl_instance_item_t substructure below.

# 7.5.2 answer_cl_instance_item_t

Variable	Description
series_t	series_t Contains the series to which this answer relates.
instance_c	uint8_t Ignore. Not used.
filler_3_s	char[3] Ignore, used for byte alignment.



## 8 CQ114 Initial Trade Reports Query

### 8.1 Query Function

This query is used for retrieving initial trade reports that have not yet been reported as delayed reports. For more information on trade reporting refer to ASX Trade Introduction and Business Information – Trade Reporting.

The initial trade reports are returned per partition and segment.

#### 8.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP7	
Struct Name	cl_query_itn_t	
Partitioned	false	
Segmented	true	
Answers	CA114	

### 8.3 Answer Properties

Transaction Type	CA114
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the
	sequence of possible structs is described below.

#### 8.4 Message Structure

### 8.4.1 cl_query_itn_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 114}.
series	series_t Fill with zeros.
instance_c	uint8_t Set to zero.
filler_1_s	char[1] Ignore. Used for byte alignment.
segment_number_n	uint16_t This field indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field as well, which indicates which segment was returned.
series_id_s	char[32] Acts as a filter on the trading code of the instrument series. Wildcard characters are supported, such as "*" matching all series or "A*" matching all series beginning with the letter A.



## 8.5 Answer Structure

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

- answer_itn_var_hdr_t
- one or more sequences of:
  - item_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_internal_asx_t (named structure 29)
      - 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)
      - cl_trade_regulatory_asx_t (named structure 35)
      - cl_trade_short_asx_t (named structure 36).

#### 8.5.1 cl_answer_itn_var_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 114}.
segment_number_n	uint16_t Indicates which segment was returned. Contains zero if this is the last segment.
items_n	uint16_t The number of top level items held in this structure.
instance_next_c	uint8_t Set to zero or one. One indicates there are further segments to read. Zero indicates this is the final (or only) answer in the segment.
filler_3_s	char[3] Ignore. Used for byte alignment.

#### 8.5.2 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.



## 8.5.3 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.

# 8.5.4 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. Users 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 Ignore. Not used.
filler_1_s	char[1] Ignore. Used for byte alignment.

## 8.5.5 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.	

## 8.5.6 cl_deal_asx_t (named structure 25)

Variable	Description
trade_slip_number_i	int32_t ASX Trade Slip number. Of the format: 1JPNNNNNC. 1 - always 1 J - last digit of day of year P - Instance number of DC



Variable	Description
	NNNNNN - Trade number/Sequence number
	C - Check digit.
exch_order_type_n	uint16_t
	Exchange specific order types.
	2 = Short Sell.
trade_condition_n	uint16_t
	The condition in which a trade was executed. Possible values include:
	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.
settlement_date_s	char[8]
	Settlement date in YYYYMMDD format.
ext_t_state_c	uint8_t
	Trade report codes. Possible value includes:
	0 = Ignore, not relevant.
	For other values refer to ASX Trade Introduction and Business Information - Trade Report
	<i>Types</i> for more information.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

# 8.5.7 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. Always set to one.
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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the trade was created. Refer to <i>ASX Trade Introduction and Business Information</i> – <i>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



Variable	Description
	Further information on the trade. Possible values include: 0 = Ignore 32 = 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.

## 8.5.8 participant_t

Variable	Description
country_id_s	char[2]
	For ASX Trade this field 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.

# 8.5.9 cl_trade_internal_asx_t (named structure 29)

Variable	Description
trading_code	trading_code_t
user_code	user_code_t See user_code_t sub structure below.
account	account_t See account_t sub structure below.
give_up_member	give_up_member_t Used to indicate the participant ID that will be clearing this trade. See give_up_member_t sub structure below.
big_attention_u	uint32_t Further information on the trade. Possible values: 0 = Ignore 32 = 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.
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.
trade_report_attribute_c	uint8_t Trade report attribute. Possible values: 0 = Initial trade report.
order_type_c	uint8_t



Variable	Description
	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.
exch_order_type_c	uint16_t
	Exchange specific order types.
	2 = Short Sell.
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.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

## 8.5.10 trading_code_t/user_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this field 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.
user_id_s	char[5] The unique identifier of the user who placed the trade. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

## 8.5.11 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.



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

## 8.5.12 give_up_member_t

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

## 8.5.13 cl_deal_trade_report_asx_t (named structure 26)

Variable	Description
extended_price_q	int64_t 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 in YYYYMMDD format.
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.

## 8.5.14 cl_deal_trade_report_asx_item_t

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

## 8.5.15 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.



Variable	Description
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.

### 8.5.16 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 <b>blank</b> 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.

## 8.5.17 cl_deal_extended_price_asx_item_t

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

## 8.5.18 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB16 when the trade involves a Tailor Made Combination (TMC) and that combination includes a mixture of equity and derivative type series. It is not included if the TMC contained only equities, or only derivatives.

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



## 8.5.19 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.

### 8.5.20 ASX Specific Overlay of regulatory_data_s Variable

All unused regulatory_da	ta_s character positio	ns are to be padded b	y 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 (default).
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.

# 8.5.21 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



## 9 CQ1134 Dedicated Trade Proxy Query

#### 9.1 Query Function

The dedicated trade query CQ1134 can be used by BSPs to recover trades for participants that are authorised in their proxy list. This query is used to regain synchronisation when a missing sequence number is detected. CQ1134 is the proxy version of CQ110.

### 9.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP7	
Struct Name	query_dedicated_trade_t	
Partitioned	True	
Segmented	False	
Answers	CA1134	

#### 9.3 Answer Properties

Transaction Type	CA1134
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.

#### 9.4 Message Structure

## 9.4.1 query_dedicated_trade_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 1134}.
series	series_t Not used in this query. Set to zeros.
instance_c	uint8_t This is the instance number of the Trade Handler process. As Trade Handler is not partitioned this value should be set to one. Sequence numbers are per day. A user should query for all trades using sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero (0) and the last trade is sent in the answer. If there are no more trades to query instance_next_c in the answer is set to zero indicating that there is no more data to query.
filler_3_s	char[3] Ignore. Used for byte alignment.
sequence_first_i	int32_t Defines the first trade of the range within which trades are sought. The sequence numbers start from one every day. If the number of items exceeds the maximum that fits into one



Variable	Description
	answer, the query should be repeated with the next missing sequence number as first argument.
sequence_last_i	int32_t Defines the last trade of the range within which trades are sought. The sequence numbers starts from one every day. A value of zero indicates that the user wishes to retrieve all available trades. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as first argument.
date_s	char[8] This must be the current date. Format: YYYYMMDD.
party	party_t The participant that the trades are queried for. See structure party_t below.

### 9.4.2 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 defines a trading participant.
	This field requires space padding.
filler_1_s	char[1] Ignore. Used for byte alignment.

#### 9.5 Answer Structure

The answer is built like a CB15 broadcast but with several trades returned. Each item in the answer represents one trade. The only time that this is not the case is when trades from an equity/derivative combination are retrieved. The trade information of the equity leg will be appended to the same CB15 broadcast as the first derivative leg. A trade consists of at least the structure cl_trade_base_api_t.

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

- answer_dedicated_trade_t
- one or more sequences of:
  - item_hdr_t
  - 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).

## 9.5.1 answer_dedicated_trade_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 110}.
instance_next_c	uint8_t Set to zero or one. Used to denote the next instance_c to query. If set to zero there is no more data to query.
filler_1_s	char[1] Ignore. Used for byte alignment.
items_n	uint16_t The number of top level items held in this structure.

# 9.5.2 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.

## 9.5.3 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



Variable	Description
	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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the deal is created. Refer to <i>ASX Trade Introduction and Business Information</i> – <i>Trade Source</i> for the possible values in this field.
open_close_req_c	uint8_t Ignore. Set to zero.
trade_type_c	uint8_t Defines the type of trade. Possible values: 1 = Standard - The trade is a normally 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. 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



Variable	Description
	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.
exchange_info_s	char[32] A free text field used at participant's discretion. For trade reports and Centre Point trades, 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 Further information on the trade. Possible values include: 0 = Ignore 32 = 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.
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.
instance_c	uint8_t Ignore. Not used.
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. Can only be a sell order (bid_or_ask_c = 2). 4 = Market Bid order (only entered by ASX Trading Operations) 8 = Price Stabilisation/Green Shoe order 64 = Centre Point order 2048 = Sweep order 4096 = Centre Point Block order.
party	party_t This is the declared counterparty 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 Ignore. Not used by ASX Trade.

# 9.5.4 trading_code_t/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 the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

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

# 9.5.6 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 defines a trading participant.
account_id_s	char[10]



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

# 9.5.7 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.

# 9.5.8 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.

# 9.5.9 cl_trade_asx_api_t (named structure 65)

Variable	Description party_t This is the declared counterparty for the other side of the trade. See party t sub structure above.		
party			
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 NNNNNN - Trade number.		
trade_condition_n	<ul> <li>uint16_t</li> <li>The condition in which a trade was executed. Possible values include:</li> <li>0 = No Condition</li> <li>2 = Internal Trade/Crossing</li> <li>8 = Buy Write (Equity/Derivative Combination).</li> <li>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.</li> </ul>		
ext_t_state_c	uint8_t Trade report codes. Possible values: 0 = Ignore, not relevant. For other values refer to ASX Trade Introduction and Business Information - Trade Report Types.		
counter_order_capacity_c	<ul> <li>c char[1]</li> <li>Dealing capacity of the counter order for crossings.</li> <li>For crossed trades, this returns the value provided in capacity_of_participant_s of the opposing order. For noncrossed</li> <li>trades, zero will be returned.</li> </ul>		

## 9.5.10 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 3 digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.		
filler_1_s	char[1] Ignore. Used for byte alignment.		

## 9.5.11 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 undefined 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]



Variable	Description	
	The date of the reported trade.	
	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 sub structure below.	

## 9.5.12 cl_deal_extended_price_asx_item_t

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

## 9.5.13 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.			

#### 9.5.14 cl_deal_trade_report_asx_item_t

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

## 9.5.15 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB15 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.

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



# 9.5.16 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.

## 9.5.17 ASX Specific Overlay of Regulatory_data_s Variable

All unused regulatory_dat	_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 (default).
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.

# 9.5.18 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



# 10 CQ1135 Dedicated Trade History, Proxy Query

## 10.1 Query Function

The dedicated trade history query CQ1135 can be used by BSPs to recover T-1 trades for participants that are authorised in their proxy list. This query works similar to CQ111 and the information is available to the participants on the following business day. CQ1135 is the proxy version of CQ111.

#### 10.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP5	
Struct Name	query_dedicated_trade_his_t	
Partitioned	True	
Segmented	False	
Answers	CA1135	

#### 10.3 Answer Properties

Transaction Type	CA1135
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.

#### 10.4 Message Structure

## 10.4.1 query_dedicated_trade_his_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'C', 'Q', 1135}.
series	series_t Not used in this query. Set to zeros.
instance_c	uint8_t This is the instance number of the Trade Handler process. As Trade Handler is not partitioned this value should be set to one. Sequence numbers are per day. A user should query for all trades using sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero (0) and the last trade is sent in the answer. If there are no more trades to query instance_next_c in the answer is set to zero indicating that there is no more data to query.
filler_3_s	char[3] Ignore. Used for byte alignment.
from_date_s	char[8] From date. Must be set to the previous trading date. Format: YYYYMMDD.
sequence_first_i	int32_t



Variable	Description
	Defines the first trade of the range within which trades are sought. The sequence numbers starts from 1 every day. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as first argument.
to_date_s	char[8] To date. Must be set to the previous trading date. Format: YYYYMMDD.
sequence_last_i	int32_t Defines the last trade of the range within which trades are sought. The sequence numbers starts from 1 every day. A value of zero indicates that the user wishes to retrieve all available trades. If the number of items exceeds the maximum that fits into one answer, the query should be repeated with the next missing sequence number as first argument.
party	party_t The participant that the trades are queried for. See party_t sub structure above.

#### 10.4.2 party_t

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

#### 10.5 Answer Structure

The answer is built like a CB15 broadcast but with several trades returned. Each item in the answer represents one trade. The only time that this is not the case is when trades from an equity/derivative combination are retrieved. The trade information of the equity leg will be appended to the same CB15 broadcast as the first derivative leg. A trade consists of at least the structure cl_trade_base_api_t.

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

- answer_dedicated_trade_his_t
- one or more sequences of:
  - item_hdr_t:
    - 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).

## 10.5.1 answer_dedicated_trade_his_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'C', 'A', 1135}.
from_date_s	char[8] From date. Format: YYYYMMDD.
sequence_first_i	int32_t Defines the first trade of the range within which trades are sought. The sequence numbers starts from one every day. If the maximum number of items for one transaction is returned, the query should be repeated with the next missing sequence number as first argument.
instance_next_c	uint8_t Set to zero or one. Used to denote the next instance_c to query. If set to zero there is no more data to query
filler_1_s	char[1] Ignore. Used for byte alignment.
items_n	uint16_t The number of top level items held in this structure.

# 10.5.2 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.

## 10.5.3 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



Variable	Description
	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 include: 1 = Bought 2 = Sold.
deal_source_c	uint8_t Where the trade is created. Refer to <i>Trade Source</i> in <i>ASX Trade Introduction and Business</i> <i>Information</i> for the possible values in this field.
open_close_req_c	uint8_t Ignore. Set to zero.
trade_type_c	uint8_t Defines the type of trade. Possible values include: 1 = Standard - The trade is 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 customer (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. 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 will be the same as created_date_s.
asof_time_s	char[6]



Variable	Description
	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 numbers are 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.
exchange_info_s	char[32] A free text field used at the 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 Further information on the trade. Possible values include: 0 = Ignore 32 = 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.
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 for more information.
trade_venue_c	uint8_t Defines the trade venue, i.e. from where the trade emanates.
instance_c	uint8_t Ignore. Not used.
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. Can only be a sell order (bid_or_ask_c = 2).



Variable	Description
	4 = Market Bid order (only entered by ASX Trading Operations)
	8 = Price Stabilisation/Green Shoe order
	64 = Centre Point order
	2048 = Sweep order
	4096 = Centre Point Block order.
party	party_t
	This is the declared counterparty for the other side of the trade.
	Refer to party_t sub structure for more information.
trade_rep_code_n	uint16_t
	Ignore. Not used by ASX Trade.
filler_2_s	char[2]
	Ignore. Used for byte alignment.
match_id	match_id_t
	Ignore. Not used by ASX Trade.

# 10.5.4 trading_code_t/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 the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

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



# 10.5.6 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.

# 10.5.7 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.

# 10.5.8 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.



# 10.5.9 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 above.
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 1OPNNNNNN: 1 - always 1 O - last digit of the Ordinal date P - Instance number of DC NNNNNN - Trade number.
trade_condition_n	uint16_t The condition in which a trade was executed. Possible values include: 0 = No Condition 2 = Internal Trade/Crossing 8 = Buy Write (Equity/Derivative Combination). This field acts as a bit mask. The binary <b>AND</b> operator can be used on the above possible values. Refer to <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u> .
ext_t_state_c	uint8_t Trade report codes. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> <i>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.

10.5.10 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 3 digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
filler_1_s	char[1] Ignore. Used for byte alignment.



## 10.5.11 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. 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. 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.

#### 10.5.12 cl_deal_extended_price_asx_item_t

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

## 10.5.13 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.

## 10.5.14 cl_deal_trade_report_asx_item_t

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



## 10.5.15 cl_combo_series_asx_t (named structure 42)

This structure will appear in the CB15 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.

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

#### 10.5.16 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.

#### 10.5.17 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 (default).
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.

## 10.5.18 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.



# **11 DQ6 Broker Signatures Query**

## 11.1 Query Function

This query returns a list of users belonging to the own participant and the instrument types/classes in which each user is permitted to trade.

#### 11.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_broker_t	
Partitioned	False	
Segmented	True	
Answers	DA6	

#### 11.3 Answer Properties

Transaction Type	DA6
Struct Name	answer_broker_t

## 11.4 Message Structure

## 11.4.1 query_broker_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 6}.
series	series_t Zeros act as a wildcard to retrieve information from all markets. Fill in country number and market code to retrieve users that have access to that particular market. Fill in a complete instrument type to retrieve information at that respective level.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field as well which indicates which segment was returned.
country_id_s	char[2] For ASX Trade this field 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_3_s	char[3] Ignore. Used for byte alignment.



## 11.5 Answer Structure

Series in the answer can specify different levels of the instrument hierarchy. The user can be allowed to trade a number of instrument types and instrument classes. For an instrument type, the series structure is completed with country_c, market_c and instrument_group_c. For an instrument class the series structure is completed with country_c, market_group_c and commodity_n. For each user, the participant ID and all legal instrument types they are authorised to trade in are returned.

## 11.5.1 answer_broker_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 6}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
country_id_s	char[2] For ASX Trade this field 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.
items_n	uint16_t The number of items held in the array.
item	answer_broker_item_t [50] Specifies all the users at the participant. See answer_broker_item_t sub structure below.

# 11.5.2 answer_broker_item_t

Variable	Description
user_id_s	char[5] The unique identifier of the user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.
program_trader_c	uint8_t Defines if the user is a Market Maker or not. Possible values include: 1 = Yes 2 = No. BO5 broadcasts are not sent for quotes (MO36 and MO37) entered by Market Maker, unless the quotes trade or are deleted.
cst_id_n	uint16_t A unique number that identifies the participant, used when subscribing for a broadcast that uses the instrument dedicated information type.
usr_id_n	uint16_t Ignore. This identifier is not used in ASX Trade.



Variable	Description
items_n	uint16_t
	The number of items held in the array.
item	unnamed item struct [100]
	Specifies the instrument types and classes in which the user is authorised to trade.
	See unnamed item sub structure below.
11.5.3 item struct for a	nswer_broker_item_t
Variable	Description
series	series_t
	Contains the instrument type or class in which the user is allowed to trade.



# 12 DQ7 Market

## 12.1 Query Function

This query is used to retrieve all markets relevant to the user.

# 12.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_market_t	
Partitioned	false	
Segmented	true	
Answers	DA7	

## 12.3 Answer Properties

Transaction Type	DA7
Struct Name	answer_market_t

## 12.4 Message Structure

## 12.4.1 query_market_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 7}.
series	series_t Acts as a filter for country and market fields only. Use a zero fill to retrieve all markets. Fill in country number to retrieve all markets for that particular exchange. Fill in country and market code to retrieve information on that particular market.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 12.5 Answer Structure

## 12.5.1 answer_market_t

The answer received contains a list of markets.

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 7}.



Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_market_item_t[100] See answer_market_item_t sub structure below.

# 12.5.2 answer_market_item_t

Variable	Description
normal_trading_days_n	uint16_t Acts as a bitmap, whereby each bit corresponds to a day in the week beginning with Monday being the lowest bit, Tuesday the next, etc. If the bit is set to one, then this indicates that the market is open for trading on that day.
normal_settl_days_n	uint16_t Acts as a bitmap, whereby each bit corresponds to a day in the week beginning with Monday being the lowest bit, Tuesday the next, etc. If the bit is set to one then this indicates that the market will be running settlement on that day.
normal_clearing_days_n	uint16_t Ignore. Not used by ASX Trade.
country_c	unint8_t An integer representing the country code.
market_c	uint8_t An integer representing the market code. For an entire list of possible values see <u>ASX</u> <u>Trade Markets, Instrument Groups and Trade Condition Codes</u> .
name_s	char[32] A character string containing the name of the market.
mar_id_s	char[5] A string type market identifier.
market_type_c	uint8_t Integer code indicating the type of market. Possible values: 1 = Stock 2 = Fixed Income 3 = Currency 4 = Power/Energy 5 = Commodity 6 = Payment 7 = Index 8 = General.
index_market_c	uint8_t Indicates if the market is an Index market. Possible values: 1 = Yes 2 = No.
bic_code_s	char[15]



Variable	Description
	Ignore. Not used by ASX Trade.
mic_code_s	char[8] Ignore. Not used by ASX Trade.
filler_2_s	char[2] Ignore. Used for byte alignment.



# 13 DQ8 Instrument Group

## 13.1 Query Function

This query is used to get information about instrument groups. For a full list of the current instrument groups refer to <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u>.

#### 13.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_instrument_group_t	
Partitioned	False	
Segmented	True	
Answers	DA8	

#### 13.3 Answer Properties

Transaction Type	DA8
Struct Name	answer_instrument_group_t

#### 13.4 Message Structure

## 13.4.1 query_instrument_group_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 8}.
series	series_t Zeros act as a wildcard. Set all fields to zero to retrieve all instrument groups. Set the country code to retrieve all instruments relating to that particular exchange. Set the country and market codes to retrieve instrument group information relating to that particular market. Fill in a complete instrument type (i.e., country, market and instrument group codes) to retrieve information on that particular instrument type (if it exists).
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 13.5 Answer Structure

The answer contains a list of instrument groups.



# 13.5.1 answer_instrument_group_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 8}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_instrument_group_item_t[100] See <i>answer_instrument_group_item_t</i> sub structure below.

# 13.5.2 answer_instrument_group_item_t

Variable	Description
extended_info_n	uint16_t
	Ignore. Currently not used.
instrument_group_c	uint8_t
	A unique binary representation of the instrument group.
name_s	char[32]
	A character string containing the name of the instrument group.
ing_id_s	char[3]
	A string type identifier of the instrument group.
group_type_c	uint8_t
	Defines the type of instrument group. Possible values include:
	1 = Option
	2 = Forward
	3 = Future
	4 = FRA
	5 = Cash
	6 = Payment
	7 = Exchange rate
	8 = Interest rate swap
	9 = REPO
	11 = Standard combination
	14 = Equity warrant.
tailor_made_c	uint8_t
	Defines whether this instrument group is used for TMCs. Possible values:
	1 = Yes
	2 = No.
option_type_s	uint8_t
	Specifies the type of option. Possible values:
	0 = Not applicable
	1 = Call
	2 = Put.
option_style_s	uint8_t



Variable	Description
	Specifies the style of option. Possible values:
	0 = Not Applicable
	1 = American
	2 = European
	3 = Asian.
warrant_c	uint8_t
	Specifies if the instrument is a warrant. Values include:
	1 = Yes
	2 = No.
average_c	uint8_t
	Ignore. Currently not used.
average_period_c	uint8_t
	Ignore. Currently not used.
repo_type_c	uint8_t
	Ignore. Currently not used.
buy_sell_back_c	uint8_t
	Ignore. Currently not used.
synthetic type c	uint8 t
, _, _	Ignore. Currently not used.
non traded ref c	uint8 t
	Ignore. Currently not used.
future styled c	uint8 t
_ / _	Specifies if the option is a future styled option. Possible values:
	1 = Yes
	2 = No.
when_issued_c	uint8_t
	Ignore. Currently not used.
is_exclusive_opening_sell_c	uint8_t
	Ignore. Currently not used.
knock variant c	uint8 t
	 Ignore. Currently not used.
binary variant c	uint8 t
	Ignore. Currently not used.
option variant c	uint8 t
option_vanant_c	Ignore. Currently not used.
physical delivery c	uint8 t
p,o.ouuo	Ignore. Currently not used.
forward style c	uint8 t
	Ignore. Currently not used.
swan style c	uint8 t
	Ignore. Currently not used.
maturity c	uint8 t
maturity_c	Ignore. Currently not used.
group short name s	char[15]
Stoup_short_name_s	cuality]



Variable	Description
	Defines a short description of the instrument group.
filler_2_s	char[2] Ignore. Used for byte alignment.



## 14 DQ17 Instrument Type

## 14.1 Query Function

This query is used to retrieve all instrument types for tradable series, including combination types.

#### 14.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_instrument_t	
Partitioned	False	
Segmented	True	
Answers	DA17	

### 14.3 Answer Properties

Transaction Type	DA17
Struct Name	answer_ext_instrument_t

#### 14.4 Message Structure

#### 14.4.1 query_instrument_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D' 'O' 17}
series	series_t Zeros act as a wildcard to retrieve all instrument types. Fill in country number and market code to retrieve instrument types related to that particular market. Fill in a complete instrument type to retrieve information on that particular instrument type.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 14.5 Answer Structure

The answer contains a list of instrument types.

A set of crossing rules can be assigned to a particular instrument type. For a brief discussion of the rules on crossing refer to *Derivative Crossings* in *ASX Trade Introduction and Business Information*.



# 14.5.1 answer_ext_instrument_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 17}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_c	uint8_t The number of items held in the array.
filler_1_s	char[1] Ignore. Used for byte alignment.
item	answer_ext_instrument_item_t[100] See answer_ext_instrument_item_t sub structure below.

# 14.5.2 answer_ext_instrument_item_t

Variable	Description
series	series_t Uniquely identifies the instrument type by the exchange, market and instrument group fields in the series_t struct. Other fields in the struct are set to zero.
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 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 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 market crossing. Whether a market has been established or not for a crossing order, the user must report the balance of 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 be lapsed for each side, beginning with the bid, and then the ask. I.e. this is the least amount of time the user 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.
min_show_vol_u	uint32_t Minimum visible volume that must be specified for iceberg orders.



Variable	Description
mar_min_time_i	int32_t Ignore. Not used.
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 Ignore. Not used.
hidden_vol_meth_n	uint16_t Hidden volume method. Possible values include: 0 = No hidden used 1 = Normal 2 = Additional.
pub_inf_id_n	uint16_t Specifies how order information is distributed. Possible values include: 1 = Without identity - The BO2 broadcast will be used. 2 = With identity - The BO1 broadcast will be used. 3 = No information (without identity) - Without identity in MQ7 and the BO2 broadcast will not be generated. 4 = No information (with identity) - Without identity in MQ7 and the BO1 broadcast will not be generated.
allow_priority_c	uint8_t Ignore. Not used.
directed_trade_information _c	uint8_t Specifies how the directed trade broadcast is disseminated. Possible values: 1 = Without counterparty 2 = With counterparty.
public_deal_information_c	uint8_t Specifies how the public trade information is distributed. Possible values include: 0 = No information 1 = Without identity 2 = With identity.
filler_1_s	char[1] Ignore. Used for byte alignment.



# 15 DQ18 Non Trading Days Query

## 15.1 Query Function

This query is used to retrieve information about which days trading and/or settlement are closed. Weekends (normally Saturdays and Sundays) are not included in the list if they are always closed.

#### 15.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_non_trading_days_t	
Partitioned	False	
Segmented	True	
Answers	DA18	

#### 15.3 Answer Properties

Transaction Type	DA18
Struct Name	answer_non_trading_days_t

#### 15.4 Message Structure

## 15.4.1 query_non_trading_days_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 18}.
series	series_t Acts as a filter for country and market fields only. Zeros act as a wildcard to retrieve information from all markets. Fill in country number and market code to retrieve information on that particular market.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 15.5 Answer Structure

#### 15.5.1 answer_non_trading_days_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 18}.



Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_non_trading_days_item_t[100] See answer_non_trading_days_t sub structure below.

# 15.5.2 answer_non_trading_days_item_t

Variable	Description
country_c	unint8_t An integer representing the country code.
market_c	uint8_t An integer representing the market code. For an entire list of possible values refer to <u>ASX</u> <u>Trade Markets, Instrument Groups and Trade Condition Codes</u> .
date_non_trading_s	char[8] A character string representing the date that is a non-normal trading day. The format of the string is YYYYMMDD.
closed_for_trading_c	uint8_t Specifies if the date is closed for trading. Possible values include: 1 = Yes 2 = No.
closed_for_settlement_c	uint8_t Specifies if the date is closed for settlement. Possible values include: 1 = Yes 2 = No.
closed_for_clearing_c	uint8_t Specifies if the date is closed for clearing. Possible values include: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.



## 16 DQ24 Exchange

## 16.1 Query Function

This query provides information on all exchanges in the system. It is currently not available for users. For ASX Trade, there is only one exchange set up in the system, that being ASX with a numerical code of 15. Users can consider this value as constant. The information for this query is left here for future reference.

#### 16.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_exchange_dq24_t	
Partitioned	False	
Segmented	True	
Answers	DA24	

#### 16.3 Answer Properties

Transaction Type	DA24
Struct Name	answer_exchange_da24_t

#### 16.4 Message Structure

## 16.4.1 query_exchange_dq24_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 24}.
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 16.5 Answer Structure

#### 16.5.1 answer_exchange_da24_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 24}.
segment_number_n	uint16_t



Variable	Description
	Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_exchange_da24_item_t[100] See answer_exchange_da24_item_t sub structure below.

# 16.5.2 answer_exchange_da24_item_t

Variable	Description
country_c	unint8_t Country/exchange identity.
	Always 15 for ASX.
opra_indicator_c	char[1] The character by which the Options Price Reporting Authority (OPRA) identifies an exchange.
name_s	char[32] The full ASCII representation of the exchange name.
exchange_short_s	char[4] Short name for exchange.
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
tz_exchange_s	char[40] The time zone environment variable for the exchange (POSIX standard). E.g. MET-1MET_DST-2,M3.5.0/2,M10.5.0/3.
master_clh_id_s	char[12] The master clearing house for the exchange.
country_s	char[2] The country ID where the exchange is located.
date_implementation_s	char[8] Implementation date. Format: YYYYMMDD.
filler_2_s	char[2] Ignore. Used for byte alignment.



# 17 DQ28 Central Group

#### 17.1 Query Function

The purpose of this transaction is to retrieve centrally defined groups. A group contains a list of series that are grouped together. These groups may indicate an alternative grouping of series that is outside the formal instrument structure. These groups are for display purposes only and have no impact on system operation.

#### 17.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	 query_central_group_t	
Partitioned	false	
Segmented	true	
Answers	DA28	

#### 17.3 Answer Properties

Transaction Type	DA28
Struct Name	answer_central_group_t

#### 17.4 Message Structure

## 17.4.1 query_central_group_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 28}.
series	series_t Acts as a filter for country and market fields to query for a particular market. Country, market and instrument group fields filter on an instrument type. Zeros act as a wildcard to retrieve information from all markets.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 17.5 Answer Structure

The answer received contains a list of series and the central group to which each series is connected.


## 17.5.1 answer_central_group_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 28}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
Item	answer_central_group_item_t[1000] List of the group name and the series in that group. See answer_central_group_item_t sub structure below.

### 17.5.2 answer_central_group_item_t

Variable	Description	
central_group_s	char[12] The ASCII representation of a centrally defined group.	
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.	
long_ins_id_s	char[32] Instrument series information that is attached to this group.	



### 18 DQ29 Session State Query

### 18.1 Query Function

This query retrieves the definition of each session state.



## Note:

To retrieve the session state of a particular item, use UQ15 or listen for the BI41 or BI741 broadcasts. For more information regarding trading session states and instrument session states refer to *ASX Trade Introduction and Business Information*.

### 18.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Came	query_trading_state_t	
Segmented	true	
Partitioned	false	
Answers	DA29	

### 18.3 Answer Properties

Transaction Type	DA29
Struct Name	answer_trading_state_t

#### 18.4 Message Structure

### 18.4.1 query_trading_state_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 29}.
Series	series_t Set to zeros.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 18.5 Answer Structure

### 18.5.1 answer_trading_state_t

Variable	Description
transaction_type	transaction_type_t



Variable	Description
	Contains the following: {'D', 'A', 29}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_trading_state_item_t[100] See answer_trading_state_item_t sub structure below.

### 18.5.2 answer_trading_state_item_t

Variable	Description
state_name_c	char[20]
	A string indicating the name of the session state, e.g. "CLOSE".
state_number_n	uint16_t
	A unique numeric identifier allocated to the session state.
iss_def_warning_interval_n	uint16_t The interval in seconds between warnings issued. Each session state transition has the ability to issue one or more warnings before it is activated. If the number of warnings is greater than zero (see the iss_def_num_of_warnings_n field below) then this time between warnings will be used.
iss_def_num_of_warnings_n	uint16_t
	The number of warnings that are issued before the session state transition is activated. The interval between warnings is given in the iss_def_warning_interval_n field above.
state_type_number_n	uint8_t
	Ignore. Not used in ASX Trade.
continues_matching_c	uint8_t Indicates if orders are automatically matched during this session state. Includes: 1 = Yes 2 = No.
trading_end_c	uint8_t Indicates if this state is the end of the trading day. Includes: 1 = Yes 2 = No.
price_quotation_required_c	uint8_t Ignore. Not used.
market_orders_allowed_c	uint8_t Ignore. Not used.
fill_or_kill_allowed_c	uint8_t Indicates if fill or kill orders are allowed during this trading state. Includes: 1 = Yes 2 = No.
fill_and_kill_allowed_c	uint8_t Indicates if fill and kill orders are allowed during this session state. Includes: 1 = Yes



Variable	Description
	2 = No.
edited_ob_changes_avail_c	uint8_t Indicates if price information broadcasts are disseminated during this session state. Includes: 1 = Yes 2 = No.
ob_changes_avail_c	uint8_t Indicates if order book changes are disseminated during this session state. Includes: 1 = Yes 2 = No.
external_full_depth_c	uint8_t Ignore. Not used.
internal_full_depth_c	uint8_t Ignore. Not used.
end_of_clearing_day_c	uint8_t Ignore. Not used.
odd_lot_allwd_c	uint8_t Ignore. Not used.
action_odd_lot_c	uint8_t Ignore. Not used.
state_priority_c	uint8_t The priority of the session state. A value between one and 255. A higher number indicates a higher priority. Used to calculate which session state takes priority when there is more than one state applicable for an instrument series or underlying.
filler_2_s	char[2] Ignore. Used for byte alignment.



### 19 DQ30 User Type Information Query

### 19.1 Query Function

This query provides information on the user's user type, legal transactions and broadcasts.

### 19.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_user_type_info_t	
Segmented	true	
Partitioned	false	
Answers	DA30	

### 19.3 Answer Properties

Transaction Type	DA30
Struct Name	answer_user_type_info_t
Segmented	true

#### 19.4 Message Structure

#### 19.4.1 query_user_type_info_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 30}.
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 19.5 Answer Structure

The answer received contains information about the user's user type, including a list of the user's legal transactions and broadcasts.

### 19.5.1 answer_user_type_info_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 30}.



Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
ust_id_s	char[5] The name of the user type.
ext_or_int_c	uint8_t If the user type is external or internal. Possible values include: 1 = External 2 = Internal.
is_trader_c	uint8_t Indicates if the user is a trader or not. Possible values include: 1 = Trader 2 = Not a trader.
program_trader_c	uint8_t Defines if the user is a Market Maker or not. Possible values include: 1 = Yes (users will only receive BO5 broadcasts for quote transactions (MO37 and MO36) when the quotes are traded or deleted). 2 = No.
trader_authorization_c	uint8_t Authorisation of the user. Possible values include: 1 = Allowed to delete and amend orders from other users in the same participant 2 = Not allowed to delete and amend orders from other users in the same participant.
filler_3_s	char[3] Ignore. Used for byte alignment.
item	answer_user_type_info_item_t[100] List of transactions and broadcasts. See answer_user_type_info_item_t sub structure below.

### 19.5.2 answer_user_type_info_item_t

Variable	Description
transaction_type	transaction_type_t Contains the message identifier of the transactions and broadcasts.
trans_or_bdx_c	uint8_t Indicates if the message is a transaction or a broadcast. Possible values include: 1 = Transaction 2 = Broadcast.
filler_3_s	char[3] Ignore. Used for byte alignment.



### 20 DQ33 Currency Query

### 20.1 Query Function

This query provides information on currencies in the system.

### 20.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_currency_t	
Segmented	True	
Partitioned	False	
Answers	DA33	

### 20.3 Answer Properties

Transaction Type	DA33
Struct Name	answer_currency_t
Segmented	True

### 20.4 Message Structure

### 20.4.1 query_currency_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 33}.
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 20.5 Answer Structure

The answer received contains the currencies in the system.

#### 20.5.1 answer_currency_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 33}.
segment_number_n	uint16_t



Variable	Description
	Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_currency_item_t[100] List of currencies. See answer_currency_item_t sub structure below.

### 20.5.2 answer_currency_item_t

Variable	Description
sec_rel_primary_n	uint16_t Relation between the primary and the secondary unit. For example, if the primary unit is <b>dollar</b> and the secondary unit is <b>cent</b> , then the relation will be 100.
third_rel_primary_n	uint16_t Relation between the primary and the tertiary unit.
base_cur_s	char[3] Defines the trading currency. This field relates to base_cur_s in ns_underlying_ basic_t struct retrieved in the DA120 response when the price_unit_c = 1 (i.e. Price).
pri_unit_s	char[15] Primary unit for the currency, e.g. <b>dollar</b> .
sec_unit_s	char[15] Secondary unit for the currency, e.g. <b>cent</b> .
third_unit_s	char[15] Tertiary unit for the currency.
pri_not_s	char[5] Primary notation for the currency. Current practice for this field is to set it to the base_cur_s value.
sec_not_s	char[5] Secondary notation for the currency. Current practice for this field is to set it to the sec_unit_s value.
third_not_s	char[5] Third notation for the currency.
acc_as_pay_c	uint8_t Accepted as payment. Possible values include: 1 = Yes 2 = No. Current practice is to set this field to two for all values as ASX Trade is a trading system only and does not include clearing.
currency_format_c	uint8_t Ignore. Currently not used.
filler_3_s	char[3] Ignore. Used for byte alignment.



### 21 DQ45 Trade Report Types Query

### 21.1 Query Function

This query is used to fetch all trade report definitions.

#### 21.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_trade_report_types_t	
Segmented	True	
Partitioned	False	
Answers	DA45	

#### 21.3 Answer Properties

Transaction Type	DA45
Struct Name	answer_trade_report_types_t

#### 21.4 Message Structure

#### 21.4.1 query_trade_report_types_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 45}.
series	series_t Set to zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 21.5 Answer Structure

The answer received contains information about trade report types.

A trade report class can be assigned to an instrument class. This indicates which trade report types are valid for that class and any series belonging to the class. See *DQ122 Delta Instrument Class*.

#### 21.5.1 answer_trade_report_types_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 45}.



Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_trade_report_types_item_t[200] A list of the trade report classes and the trade report types within the class. See answer_trade_report_types_item_t sub structure below.

### 21.5.2 answer_trade_report_types_item_t

Variable	Description
initial_trr_min_value_u	int64_t The required minimum order value for initial trade reports.
trc_id_s	char[10] The ID string for a trade report class. The trade report class contains a list of trade report types.
trr_id_s	char[4] The ID string for a trade report type.
condition_s	char[32] The description of the trade report type.
authorized_c	uin8t_t Is the user authorised for this trade report type. Possible values include: 1 = Yes 2 = No.
ext_t_state_c	uint8_t Trade report code. Refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business Information</i> for more information.
allow_interbank_c	uint8_t The trade report type is allowed to report between different participants. Possible values include: 1 = Yes 2 = No.
allow_within_participant_c	uint8_t The trade report type is allowed to report within the same participant (i.e. a crossing). Possible values include: 1 = Yes 2 = No.
cbo_trade_report_c	uint8_t Describes if the Trade Report Type is used to do a combo trade report. Possible values include: 1 = Yes 2 = No.



Variable	Description
allow_non_std_settlement_ c	uint8_t Allow a non-standard settlement date in the trade report. Possible values include: 1 = Yes 2 = No.
time_of_agree_req_c	uint8_t Specifies how time of agreement is specified and validated in the trade report. Possible values include: 0 = Not required 1 = Required on first reported side 2 = Required on both sides - not matched 3 = Required on both sides - must match. Always set to 3 on ASX Trade.
time_of_agree_gran_c	uint8_t Specifies if the time of agreement contains only a date or both date and time. Possible values include: 0 = Not applicable 1 = Date 2 = Date and time. Always set to 1 on ASX Trade.
filler_2_s	char[2] Ignore. Used for byte alignment.



### 22 DQ46 Deal Source Query

### 22.1 Query Function

This query is used to retrieve all available deal sources. The deal source refers to where a trade is created.

#### 22.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_deal_source_t	
Segmented	True	
Partitioned	False	
Answers	DA46	

### 22.3 Answer Properties

Transaction Type	DA46
Struct Name	answer_deal_source_t
Segmented	True

#### 22.4 Message Structure

#### 22.4.1 query_deal_source_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 46}.
series	series_t Set to zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 22.5 Answer Structure

The answer received contains a list of all available deal sources. Each response is prefaced with the transaction type (DA46).

### 22.5.1 answer_deal_source_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 46}.



Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_deal_source_item_t[100] See answer_deal_source_item_t sub structure below.

## 22.5.2 answer_deal_source_item_t

Variable	Description
ds_attribute_q	int64_t
	Defines the attribute of the deal source. Different behaviours may be controlled by the
	attribute. This field is a bitmap. Possible values include:
	0 = Unassigned
	bit 1 = Trade report
	bit 2 = Bulletin board
	bit 3 = Excluded from trade statistics
	bit 4 = Outside exchange
	bit 5 = Filtered public trade broadcast
	bit 6 = Centre Point match.
deal_source_n	uint16_t
	Unique identifier for this deal source.
desc_long_s	char[128]
	A textual description for the deal source. May contain blanks.
desc_abbreviated_s	char[32]
	An abbreviated textual description for the deal source. May contain blanks.
filler_2_s	char[2]
	Ignore. Used for byte alignment.



### 23 DQ53 Corporate Action Query

### 23.1 Query Function

This query returns all active corporate action codes that exist for a linked underlying, an underlying, an instrument class or an instrument series.

#### 23.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_corp_action_t	
Segmented	True	
Partitioned	False	
Answers	DA53	

#### 23.3 Answer Properties

Transaction Type	DA53
Struct Name	answer_corp_action_da53_t

#### 23.4 Message Structure

#### 23.4.1 query_corp_action_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 53}.
series	series_t Defines for which corporate action codes the query is searching. Set to zeros for information for instruments from all markets. Fill in country_c and market_c to retrieve information for instruments belonging to that market. Fill in country_c, market_c and instrument_group_c to retrieve information for all instruments belonging to that instrument type.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 23.5 Answer Structure

The answer received contains a list of linked underlyings, underlyings, instrument classes or instrument series and their associated corporate action code(s). The corp_level_action_c field in the answer indicates on which level the code is assigned.



Each linked underlying, underlying, instrument class or instrument series could have several entries in the answer, depending on how many codes it has assigned.

An instrument series always inherits all codes assigned on a higher level (linked underlying, underlying or instrument class). This means that for one instrument series the same code can be assigned several times.

### 23.5.1 answer_corp_action_da53

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 53}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_corp_action_da53_item_t[1000] See answer_corp_action_da53_item_t sub structure below.

### 23.5.2 answer_corp_action_da53_item_t

Variable	Description
series	<ul> <li>series_t</li> <li>This is filled in with different information depending on the level the corporate action code is assigned to:</li> <li>Linked Underlying level</li> <li>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 DQ120 for more information.</li> <li>Underlying level</li> <li>Only a value in commodity_n is filled in, the rest of the fields are zero.</li> <li>Instrument Class level</li> <li>A value in country_c, market_c, instrument_group_c and commodity_n.</li> <li>Instrument Series level</li> <li>A complete series definition.</li> </ul>
corp_action_code_s	char[2] Corporate Action Code.
corp_action_type_c	uint8_t Indicates if the code is a corporate action or a notification code. Possible values include: 1 = Basis of Quotation 2 = Status Note.
corp_action_status_c	uint8_t It is possible to disable a corporate action or notification code during the day. However this response will only return "Enabled" corporate actions. Therefore this field will always return one.
corp_action_level_c	uint8_t The instrument level to which the corporate action code has been assigned. Possible values include:



Variable	Description
	1 = Underlying
	2 = Linked Underlying
	3 = Instrument Class
	4 = Instrument Series.
filler_3_s	char[3]
	Ignore. Used for byte alignment.



### 24 DQ55 Clearing Participants Query

#### 24.1 Query Function

This query is used to receive the list of valid clearing participants for a participant.

Valid clearing participants must be used in order and trade transactions to indicate through which clearer or cost centre the transaction is to be processed. In the order and trade transactions the clearing ID is entered into the ex_customer_s field in the give_up_member_t structure.

# (j)

Note:

The clearing_customer_s field in the answer is of type char[12] whereas the ex_customer_s field in the give_up_member_t structure is only char[5].

ASX Trade will only use a single character clearing ID, so this should be placed in the left most character of the ex_customer_s field. The rest of the field must be space padded.

#### 24.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_cl_participant_t	
Segmented	True	
Partitioned	False	
Answers	DA55	

#### 24.3 Answer Properties

Transaction Type	DA55
Struct Name	answer_cl_participant_t

#### 24.4 Message Structure

#### 24.4.1 query_cl_participant_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 55}.
series	series_t Set to zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.



### 24.5 Answer Structure

## 24.5.1 answer_cl_participant_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 55}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_cl_participant_item_t[200] List of valid clearing IDs for the participant. See answer_cl_participant_item_t sub structure below.

### 24.5.2 answer_cl_participant_item_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.
clearing_customer_s	char[12] Clearing participant identifier. The value found in this field is to be used in the give_up_member_t struct, in the ex_customer_s field. Users will note that the two fields are of different length – this one being 12 char and the ex_customer_s field being five char. However, the value itself will typically be a single digit number. The single digit needs to be used in the ex_customer_s field and the rest of the field space padded.
name_s	char[32] Description of the clearing identity.
default_customer_c	uint8_t Indicates that this is the default clearing participant for the participant. 1 = Yes 2 = No.



### 25 DQ56 Special Liability Query

#### 25.1 Query Function

This query is used to retrieve all the special liability limits for a user.

Special liability limits can be set for each market by ASX Trading Operations on the advice from the participant compliance officer. Once set, users will not be permitted to enter orders that exceed the value specified by the special liability.

#### 25.2 Query Properties

omniapi_query_ex	
EPO	
query_special_liability_t	
True	
False	
DA56	
	omniapi_query_ex         EP0         query_special_liability_t         True         False         DA56

#### 25.3 Answer Properties

Transaction Type	DA56
Struct Name	answer_special_liability_t

#### 25.4 Message Structure

### 25.4.1 query_special_liability_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 56}.
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 25.5 Answer Structure

The answer received contains a list of markets and their limits connected to the user.

### $25.5.1 \quad answer_special_liability_t$

Variable	Description
transaction_type	transaction_type_t



Variable	Description
	Contains the following: {'D', 'A', 56}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
trading_code	trading_code_t See trading_code_t sub structure below.
item	answer_special_liability_item_t[200] List of valid clearing IDs for the participant. See answer_special_liability_item_t sub structure below.

## 25.5.2 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 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 the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

### 25.5.3 answer_special_liability_item_t

If no item is returned for a particular market then the user can assume unlimited liability has been given for that market.

Variable	Description
spec_liab_limit_i	int32_t Special liability limit for each market. Zero means unlimited.
country_c	uint8_t Country and exchange identity. Always 15 for ASX.
market_c	uint8_t An integer representing the market code. For valid market codes see <u>ASX Trade Markets,</u> Instrument Groups and Trade Condition Codes.
filler_2_s	char[2] Ignore. Used for byte alignment.



### 26 DQ57 Member Obligation Query

#### 26.1 Query Function

This query is used to receive a list of participants the user has the rights to receive trading participant specific information on behalf of. This is used by BSPs to query who they are permissioned to received dedicated order and trade information from.

The information returned in the cst_id_n_i field can be used when subscribing for the BO5 broadcast for a particular participant.

#### 26.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	query_member_obligation_t
Segmented	True
Partitioned	False
Answers	DA57

### 26.3 Answer Properties

Transaction Type	DA57
Struct Name	answer_member_obligation_da57_t

#### 26.4 Message Structure

#### 26.4.1 query_member_obligation_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 57}.
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 26.5 Answer Structure

#### 26.5.1 answer_member_obligation_da57_t

Variable	Description
transaction_type	transaction_type_t



Variable	Description
	Contains the following: {'D', 'A', 57}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_member_obligation_da57_item_t[500] See answer_member_obligation_da57_item_t sub structure below.

### 26.5.2 answer_member_obligation_da57_item_t

Variable	Description
series	series_t The market to which the obligation relates.
cst_id_n_i	uint16_t A unique number that identifies the participant, used when subscribing for directed broadcast information.
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.
right_type_c	uint8_t The rights for the participant. Possible values include: 0 = No rights 1 = The user doing the query is able to act on behalf of the member indicated in the cst_id_n_i field.
filler_2_s	char[2] Ignore. Used for byte alignment.



### 27 DQ87 Market Maker Protection Query

#### 27.1 Query Function

This query is used to provide information of the market maker protection parameters defined for the participant and underlying. It's only applicable to market maker users.

#### 27.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_mm_protection	
Segmented	True	
Partitioned	False	
Answers	DA87	

#### 27.3 Answer Properties

Transaction Type	DA87
Struct Name	answer_mm_protection

#### 27.4 Message Structure

#### 27.4.1 query_mm_protection

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D' 'O' 87}
series	series_t Must be zeros.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 27.5 Answer Structure

#### 27.5.1 answer_mm_protection

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 87}.
segment_number_n	uint16_t



Variable	Description
	Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_mm_protection_t[500] See answer_mm_protection_t sub structure below.

### 27.5.2 answer_mm_protection_item_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 the quotes for the instruments connected to the underlying. A value of 0 means that no quantity protection exists.
delta_protection_q	Int64_t When changing quantities there are two options: delta and absolute. Delta changes amend the quantity/total volume of an order by the given amount, positive to increase the quantity, negative to reduce the quantity. Absolute changes means that the quantity/total volume should be set to the value in the quantity/total volume field.
exposure_time_interval_i	Int32_t Specifies the rolling time interval in milliseconds used in quantity/delta protection calculations.
frozen_time_i	Int32_t Specifies the time interval in milliseconds when quotes are rejected after Market Maker protection has been triggered.
Commodity_n	UInt16_t Underlying definitions are defined by each exchange. Commodity Code is a part of the Series definition.
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.
includes_futures_c	UInt8_t Indicates if the market is an index market or not. Possible values: 1 = Yes 2 = No
filler_2_s	char[2] Ignore. Used for byte alignment.



### 28 DQ120 Delta Underlying Query

#### 28.1 Query Function

This query is used to retrieve information about a new underlying or an underlying that has changed. To reduce network load and improve application data initialisation times, this query uses the delta query technique that allows users to retrieve only those changes since they last received underlying data, rather than always requesting all the underlyings in the system. This technique is fully explained in *Delta Queries* in *ASX Trade Introduction and Business Information*.

#### 28.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_delta_t	
Segmented	True	
Partitioned	False	
Answers	DA120	

#### 28.3 Answer Properties

Transaction Type	DA120
Struct Name	This is a Variable Information Answer (VIA). The first structure is the header indicating that it is a DA120, how many items it has, their total size and the segment number. This is followed by a series of item structures, each with their own sub header. The application is required to check the item's sub header in order to know what type of item follows it. Unknown structures should simply be ignored.

#### 28.4 Related Messages

Related messages include:

- Related queries DQ122, DQ124, DQ126
- Related broadcasts BU120, BU122, BU124, BU126.

#### 28.5 Message Structure

#### 28.5.1 query_delta_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 120}.
series	series_t Set to zero.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.



Variable	Description
filler_2_s	char[2] Ignore. Used for byte alignment.
download_ref_number_q	int64_t Reference number to use in delta queries and answers. Each time a change is made to an underlying in ASX Trade, this number is incremented. It can then be used to retrieve all changes to underlyings since a particular reference point. To receive the delta, increment the latest received number from the answer of this query or the latest broadcast related to the query. To enforce a full answer use -1 in this field.
full_answer_timestamp	time_spec_t Set this structure to the timestamp of the last full answer received. To enforce a full answer set to zero. See time_spec_t sub structure below.

### 28.5.2 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.

#### 28.6 Answer Structure

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

- answer_segment_hdr_t
- one of:
  - item_hdr_t
  - sub_item_hdr_t
  - 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).



### 28.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 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 Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 28.6.2 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.	

### 28.6.3 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.

## 28.6.4 ns_delta_header_t (named struct number 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.
full_answer_timestamp	time_spec_t See time_spec_t sub structure above.
full_answer_c	uint8_t Indicates if a full answer has been enforced in the delta query. Possible values include: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.



### 28.6.5 ns_remove_t (named structure 37002)

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



Note:

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

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

#### 28.6.6 ns_underlying_basic_t (named structure 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. There is no guarantee when retrieving a full answer that an underlying will be received by the user before any others that are linked to it through this field. Any linked underlyings that do not have an associated instrument class will not be disseminated.
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 for more information</i> . 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. The International Securities Identification Number (ISIN) shall consist of: 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.
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).



Variable	Description
	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	uint8_t
	The type of underlying. Possible values include:
	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
	$10 = 2 \text{ for } R_{\text{M}}$ and $\Delta gricultures Index$
	12 = Metal Index.
nrice unit c	uint8 t
price_unit_e	The price unit for the underlying Possible values include:
	1 – Drica
	2 - Deinte
	5 - Folits
	5 - Ilvivi Illuex
	$\theta = Basis Points$
	8 = Percentage of Nominal
	9 = Dirty Price.
underlying_status_c	uint8_t
	Possible values include:
	1 = Active
	2 = Delisted.
	It is possible for the DA120 to return delisted underlyings. The current ASX Trading
	Operations procedure is to leave underlyings in a delisted state for a period of time before
	manually deleting them.
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	
	For ASX Trade this is always set to "AU", indicating the Australian exchange.
ext_provider_c	char[1]



Variable	Description
	External Price feed provider.
	N = NMF
	S = Six
	O = Six OMX
	F = Direct Feed
	R = Direct Feed OPRA
	T = Transaction
	L = LMIL
	E = Reuter SSL.
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: Users 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 ns_remove_t (named structure 37002).
	Possible values include:
	1 = Insert
	2 = Update.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

### 28.6.7 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 instruments.
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



Variable	Description
	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.
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 = Bon 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.

## $28.6.8 \quad ns_coupon_dates_t-named \ struct\ number\ 37203$

Variable	Description
date_coupdiv_s	char[8] Coupon date for bond underlying or dividend date for stock underlying. Format: YYYYMMDD.



Variable	Description
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.



### 29 DQ122 Delta Instrument Class Query

#### 29.1 Query Function

This query is used to retrieve information about instrument classes.

To reduce network load and improve application data initialisation times, this query uses the delta query technique that allows users to retrieve only those changes since they last received class data, rather than always requesting all the classes in the system. This technique is fully explained in *Delta Queries* in *ASX Trade Introduction and Business Information*.

Users should note that a full answer will return only those classes denoted as traded (traded_c field set to one). A delta response however will return any changes to instrument classes, which includes ones that are no longer traded.

#### 29.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_delta_t	
Segmented	True	
Partitioned	False	
Answers	DA122	

#### 29.3 Answer Properties

Transaction Type	DA122
Struct Name	The message complies with the VIM concept and has no topmost struct. The first structure is the header indicating that it is a DA122, how many items it has, their total size and the segment number. This is followed by a series of item structures, each with their own sub- header. The application is required to check the item's sub-header in order to know what type of item follows it. Unknown structures should simply be ignored.

#### 29.4 Related Messages

Related messages include:

- Related queries DQ120, DQ124, DQ126
- Related broadcasts BU120, BU122, BU124, BU126.

#### 29.5 Message Structure

#### 29.5.1 query_delta_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 122}.
series	series_t



Variable	Description
	Acts as a filter for country and market fields only. Zeros act as a wildcard to retrieve information from all markets. Fill in country number and market code to retrieve information on that particular market.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.
download_ref_number_q	<ul> <li>int64_t</li> <li>Reference number to use in delta queries and answers.</li> <li>Each time a change is made to a class in ASX Trade this number is incremented. It can then be used to retrieve all changes to classes since a particular reference point.</li> <li>To receive the delta, increment the latest received number from the answer of this query or the latest broadcast related to the query. To enforce a full answer use -1 in this field.</li> </ul>
full_answer_timestamp	time_spec_t Set this structure to the timestamp of the last full answer received. To enforce a full answer set to zero. See time_spec_t sub structure below.

#### 29.5.2 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.

### 29.6 Answer Structure

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

- answer_segment_hdr_t
- one of:
  - item_hdr_t:
    - sub_item_hdr_t
      - 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_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).

### 29.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 122}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the message, including this header.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 29.6.2 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.

#### 29.6.3 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.

## 29.6.4 ns_delta_header_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.
full_answer_timestamp	time_spec_t See time_spec_t sub structure above.
full_answer_c	uint8_t



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

### 29.6.5 ns_remove_t (named structure 37002)

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



Note:

Users 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. Users need to cater for this situation.

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

#### 29.6.6 ns_inst_class_basic_t (named structure 37101)

Variable	Description
series	series_t Contains the instrument class to which this information relates.
upper_level_series	upper_level_series_t (same as 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 derivative (i.e. where derivate_level_n > 0). E.g. for an option on a stock, this field will contain the instrument class of the stock. There is no guarantee when retrieving a full answer that an instrument class will be received by the user before any others that are linked to it through this field.
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 x 100 = 650 cents.
contract_size_i	<ul> <li>int32_t</li> <li>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 x 100 = 1,000 of the underlying security.</li> <li>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.</li> <li>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.</li> </ul>


Variable	Description
redemption_value_i	int32_t Redemption value represents the amount paid at maturity. The redemption value will be equal to the nominal value except for securities with amortisation or options. The redemption value is expressed as a percentage of the nominal value. The value is a decimal value stored with six decimals, e.g. 100% is stored as 1000000.
undisclosed_min_ord_val_i	int32_t Minimum order value for undisclosed quantity orders. The value is always expressed in the primary currency unit. The value is defined as quantity * price * price quotation factor. Assumes decimal places have not been applied.
opt_min_ord_val_i	int32_t Optional minimum order value. The value is always expressed in the primary currency unit. The value is defined as quantity * price * price quotation factor. Assumes decimal places have not been applied.
opt_min_trade_val_i	int32_t Optional minimum trade value. The value is always expressed in the primary currency unit. The value is defined as quantity * price * price quotation factor. Assumes decimal places have not been applied.
derivate_level_n	uint16_t The derivative level of the instrument. Possible values include: 0 = Stock 1 = Derivative on a stock 2 = Derivative on a derivative. E.g., an option on a future of a stock.
dec_in_strike_price_n	uint16_t Number of implicit decimals in the strike_price_i field (part of series_t structure). A value of zero means no decimals.
dec_in_contr_size_n	uint16_t Number of implicit decimals in the contract_size_i and the price_quot_factor_i fields. A value of zero means no decimals.
rnt_id_n	uint16_t This identifies how the instrument is ranked. Always set to one.
virt_commodity_n	uint16_t 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. If zero, no virtual underlying is used but the real underlying code is used in broadcast subscriptions.
settlement_days_n	uint16_t 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 include: 0 = Not applicable



Variable	Description
	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 include: 1 = Yes 2 = No.
price_unit_premium_c	uint8_t The premium unit for the class. Possible values include: 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 include: 1 = Price 2 = Yield 3 = Points 6 = Basis points 7 = Inverted yield.
indicative_prices_c	uint8_t Indicative prices. Possible values include: 1 = Yes 2 = No.
trd_cur_unit_c	uint8_t Specifies the currency unit the instrument is traded in. Possible values include: 1 = Primary unit 2 = Secondary unit 3 = Tertiary unit.



Variable	Description
	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	<ul> <li>uint8_t</li> <li>Operation that the trade system has carried out on this particular item.</li> <li>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).</li> <li>Possible values include:</li> <li>1 = Insert</li> <li>2 = Update.</li> </ul>
csd_id_s	char[12] Ignore. Not used by ASX Trade.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 29.6.7 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	tick_size_t See tick_size_t sub structure below.
dec_in_premium_n	uint16_t Number of implicit decimals in the premium/price.
is_fractions_c	char[1] Indicates if the premium is represented as fractions. Always set to "N" meaning "No".
price_format_c	uint8_t Premium/price format. Always set to zero.

# 29.6.8 tick_size_t

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



# 29.6.9 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 zero for all instrument types, indicating no limit.	
block_n	uint32_t	
	Defines the block size of the class.	
	Currently set to one for all instrument types.	
lot_type_c	uint8_t	
	Type of lot. Possible values include:	
	1 = Odd lot	
	2 = Round lot	
	3 = Block lot	
	4 = All or none.	
	Currently set to two for all instrument types.	
filler_3_s	char[3]	
	Ignore. Used for byte alignment.	

# 29.6.10 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 include:
	1 = Up
	2 = Down.
clean_pr_ud_c	uint8_t
	Clean price up/down. Possible values include:
	1 = Up
	2 = Down.
day_count_conv_c	uint8_t
	Day count convention. Possible values include:
	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 include:
	1 = SAME
	2 = LAST360
	3 = LAST.
set_start_consid_c	uint8_t
	Start consideration. Possible values include:
	1 = Yes
	2 = No.
set_end_consid_c	uint8_t
	End consideration. Possible values include:
	1 = Yes
	2 = No.
calculation_conv_c	uint8_t
	Calculation convention. Possible values include:
	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_1_s	char[3]
	Ignore. Used for byte alignment.

## 29.6.11 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 query's answer if the crossing rule from the associated instrument type is not applicable for this class. For a brief discussion of the rules on crossing, see *Derivative Crossings* in *ASX Trade Introduction and Business Information*.

ASX Trade has implemented crossing rules at the instrument type level. 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 be lapsed for each side, beginning with the bid, then the ask. I.e. this is the least amount of time the user 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 include: 1 = Yes 2 = No.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 29.6.12 ns_inst_class_ext1_t (named structure 37107)

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



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



# 30 DQ124 Delta Instrument Series

### 30.1 Query Function

This query is used to retrieve information about an instrument series.

To reduce network load and improve application data initialisation times, this query uses the delta query technique that allows users to retrieve only those changes since they last received series data, rather than always requesting all series in the system. This technique is fully explained in *Delta Queries* in *ASX Trade Introduction and Business Information*.

Users should note that a full answer will return only those series denoted as traded (traded_in_click_c field set to 1) and have their last traded date set in the future (see date_last_trading_s field). A delta response however, will return any changes to instrument series, which includes ones that are no longer traded, ones that are suspended or ones that have a last traded date in the past.

### 30.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_delta_t	
Segmented	True	
Partitioned	False	
Answers	DA124	

### 30.3 Answer Properties

Transaction Type	DA124
Struct Name	The message complies with the VIM concept and has no topmost struct. The first structure is the header indicating that it is a DA124, how many items it has, their total size and the segment number. This is followed by a series of item structures, each with their own sub- header. The application is required to check the item's sub-header in order to know what type of item follows it. Unknown structures should be ignored.

### 30.4 Related Messages

Related messages include:

- Related queries: DQ120, DQ122, DQ126
- Related broadcasts: BU120, BU122, BU124, BU126.

## 30.5 Message Structure

#### 30.5.1 query_delta_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 124}.
series	series_t



Variable	Description
	Zeros act as a wildcard to retrieve all instrument series. Fill in country number and market code to retrieve instrument series related to that particular market. Fill in a complete instrument type to retrieve series matching that particular instrument type. Fill in a complete instrument class or series to retrieve a series matching those values. If the filtering does not match any series then only the ns_delta_header_t will be returned in the response.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.
download_ref_number_q	<ul> <li>int64_t</li> <li>Reference number to use in delta queries and answers.</li> <li>Each time a change is made to a series in ASX Trade this number is incremented. It can then be used to retrieve all changes to series since a particular reference point.</li> <li>To receive the delta, increment the latest received number from the answer of this query or the latest broadcast related to the query. To enforce a full answer use minus one in this field.</li> </ul>
full_answer_timestamp	time_spec_t Set this structure to the timestamp of the last full answer received. To enforce a full answer set to zero. See time_spec_t sub structure below.

# 30.5.2 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.

## **30.6** Answer Structure

This is a variable information answer. Headers and sub headers within the message identify what is contained in the message.

The overall structure is:

- answer_segment_hdr_t
- one of:
  - item_hdr_t
    - sub_item_hdr_t
      - 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 both of the following:
  - ns_inst_series_basic_t (named structure 37301)
  - ns_inst_series_basic_single_t (named structure 37302).

# 30.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 124}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the message, including this header.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 30.6.2 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

# 30.6.3 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.

# 30.6.4 ns_delta_header_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.
full_answer_timestamp	time_spec_t See time_spec_t sub structure above.



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

# 30.6.5 ns_remove_t (named structure 37002)

This indicates that the series has been removed. Only this structure will be sent if the series has been removed. A zero value on the strike price but not on the expiry date indicates a wildcard, meaning that all instrument series matching the country, market, group, class and expiry date should be deleted. This is typically seen when a set of derivatives expire all on the same day. Users should note here that it is possible for a warrant over an index to have a strike price of zero.



## Note:

Users 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. Users need to cater for this situation.

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

### 30.6.6 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 expirations, the value in this field will be 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]



Variable	Description
	The last valid trading date of the series in YYYYMMDD format. The date here, together with 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 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	<ul> <li>uint8_t</li> <li>The actual state of the series. Possible values include:</li> <li>1 = Active (both expired and not expired)</li> <li>2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status)</li> <li>3 = Issued (series exists in the system but is unavailable for trading - to the participant this is the same as delisted)</li> <li>4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).</li> </ul>
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values include: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values include: 1 = Yes 2 = No.
db_operation_c	uint8_t Operation that ASX Trade has carried out on this particular item. <b>Note</b> : Users 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 user. As mentioned above, a remove will be indicated by the named structure ns_remove_t (named structure 37002). Possible values include: 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.



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

# 30.6.7 ns_inst_series_basic_single_t (named structure 37302)

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

Variable	Description
upper_level_series	series_t This field will only contain information if the series to which this whole structure relates is a second level derivate i.e., if the series is a derivative of a derivative. Derivate 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 user before any others that are linked to it through this field.
contract_size_i	<ul> <li>int32_t</li> <li>For an option, this indicates the contract size, i.e., the number of units of the underlying security that concern 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 x 100 = 1,000 of the underlying security.</li> <li>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.</li> <li>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.</li> <li>This field will only have a value if it differs from the associated instrument class.</li> </ul>
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	<ul> <li>uint16_t</li> <li>Indicates the trading state of this instrument series. This field is only used when the series is first created, indicating its initial instrument session state. Users should note that this may not be the active state of the instrument. Refer to <i>Trading Session, Instrument Session and Active Session States</i> in <i>ASX Trade Introduction and Business Information</i> to determine how to find the active state for an instrument series.</li> <li>Available values can be retrieved through the Trading State Query (refer to <i>DQ29 Trading State</i> for more information).</li> </ul>
ex_coupon_n	uint16_t Ex coupon period.
isin_code_s	char[12] A code that uniquely identifies an ISIN. The ISIN consists of: 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]



Variable	Description
	Ignore. Not used.
first_settlement_date_s	char[8] First settlement date in YYYYMMDD format.
date_notation_s	char[8] Notation date in YYYYMMDD format.
deliverable_c	uint8_t Defines if a series can be delivered or not (cash settlement). Possible values include: 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.



# 31 DQ126 Combination Series

# 31.1 Query Function

This query is used to retrieve information about combination series.

# 31.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_combo_t	
Segmented	True	
Partitioned	False	
Answers	DA126	

# 31.3 Answer Properties

Transaction Type	DA126
Struct Name	The message complies with the VIM concept and has no topmost struct. The first structure is the header indicating that it is a DA124, how many items it has, their total size and the segment number. This is followed by a series of item structures, each with their own sub- header. The application is required to check the item's sub-header in order to know what type of item follows it. Unknown structures should simply be ignored.

# 31.4 Related Messages

Related messages include:

- Related queries: DQ120, DQ122, DQ124
- Related broadcasts: BU120, BU122, BU124, BU126.

# 31.5 Message Structure

# 31.5.1 query_combo_t

Variable	Description
transaction_type	transaction_type_t
	Set the structure to the following: {'D', 'Q', 126}.
series	series_t
	Set to zero for information from all markets.
	Fill in country_c and market_c to retrieve information on that market.
	Fill in country_c, market_c and instrument_group_c to retrieve information on that
	instrument type.
segment_number_n	uint16_t
	Indicates which segment users wish to receive, typically starting at one. The reply structure
	has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2]



Variable	Description
	Ignore. Used for byte alignment.

## 31.6 Answer 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:

- answer_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).

# 31.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t
	Contains the following: { D , A , 126}.
items_n	uint16_t
	The number of top level items following this header.
size_n	uint16_t
	The size of the message, including this header.
segment_number_n	uint16_t
	Indicates the segment for this particular reply. A value of zero means that there are no
	more segments.
filler_2_s	char[2]
	Ignore. Used for byte alignment.

# 31.6.2 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

# 31.6.3 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.



# 31.6.4 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 expirations, the value in this field will be 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 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 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	<ul> <li>uint8_t</li> <li>The actual state of the series. Possible values include:</li> <li>1 = Active (both expired and not expired)</li> <li>2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status)</li> <li>3 = Issued (series exists in the system but is unavailable for trading 0 - to the user this is the same as delisted)</li> <li>4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).</li> </ul>
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values include: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values include: 1 = Yes



Variable	Description
	2 = No.
db_operation_c	<ul> <li>uint8_t</li> <li>Operation that the trade system has carried out on this particular item.</li> <li><b>Note</b>: Users 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 user. As mentioned above, a remove will be indicated by the named structure ns_remove_t (named structure 37002).</li> <li>Possible values include:</li> <li>1 = Insert</li> <li>2 = Update.</li> </ul>
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.

# 31.6.5 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
	include:
	B = Buy
	S = Sell.
op_if_sell_c	char[1]
	Specifies whether to buy or sell the series when selling the combination. Possible values
	include:
	B = Buy
	S = Sell.



# 32 DQ127 Historical Series

### 32.1 Query Function

This query is used to retrieve information about expired single instrument series. It is used by OI applications when viewing historical orders for expired single instrument series. Expired instrument series are not available from the answer to a Delta Instrument Series Query (refer to DQ124 Delta Instrument Series for more information).

This query involves a date field along with a list of unknown series. The response supplies information on each series given in the list as of the date supplied in the query.

### 32.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	query_history_t
Segmented	true
Partitioned	false
Answers	DA127

## 32.3 Answer Properties

Transaction Type	DA127
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.

## 32.4 Related Messages

A related message includes the related query - DQ128.

### 32.5 Message Structure

## 32.5.1 query_history_t

The query contains an item list where all the unknown series are sent.

Variable	Description
transaction_type	transaction_type_t
	Set the structure to the following: {'D', 'Q', 127}.
series	series_t Set to zeros for information from all markets. Fill in country_c and market_c to retrieve information on that market.
	instrument type.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.



Variable	Description
items_n	uint16_t The number of unknown instrument series held in the array
date_trading_s	char[8] The date from which the historical series should be picked from. Format: YYYYMMDD.
item	query_history_item_t[200] List of unknown instrument series. See query_history_item_t sub structure below.

# 32.5.2 query_history_item_t

Variable	Description
series	series_t Identifies the unknown instrument series.

## **32.6** Answer Structure

This is a variable information answer. Headers and sub headers within the message identify what is contained in the message.

The overall structure is:

- answer_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_inst_series_basic_single_t (named structure 37302).

## 32.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 127}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the message, including this header.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2] Ignore. Used for byte alignment.



# 32.6.2 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.

### 32.6.3 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.

# 32.6.4 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 expirations, the value in this field will be 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 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 in YYYYMMDD format. This date here, together with the time_first_trading_s field provides the first trading time in UTC.



Variable	Description
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	<ul> <li>uint8_t</li> <li>The actual state of the series. Possible values include:</li> <li>1 = Active (both expired and not expired)</li> <li>2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status)</li> <li>3 = Issued (series exists in the system but is unavailable for trading - to the user this is the same as delisted)</li> <li>4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).</li> </ul>
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values include: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values include: 1 = Yes 2 = No.
db_operation_c	uint8_t Operation that ASX Trade has carried out on this particular item. <b>Note</b> : Users 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 user. As mentioned above, a remove will be indicated by the named structure ns_remove_t (named structure 37002). Possible values include: 1 = Insert 2 = Update.
trade_reporting_only_c	uint8_t Ignore, not used. Always set to 2.
filler_2_s	Char[2] Ignore. Used for byte alignment.

# 32.6.5 ns_inst_series_basic_single_t (named structure 37302)

Both this structure and ns_inst_series_basic_t (named structure 37301) are sent for each expired series.

Variable	Description
upper_level_series	series_t This field will only contain information if the series to which this whole structure relates 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 user before any others that are linked to it through this field.
contract_size_i	int32_t



Variable	Description
	<ul> <li>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 x 100 = 1,000 of the underlying security.</li> <li>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.</li> <li>The exceptions to this rule are for index warrants where it represents the cash value per</li> </ul>
	index point and for basket warrants where it represents the minimum parcel of warrants required to be held for early exercise. This field will only have 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 for an expired series.
ex_coupon_n	uint16_t Ex coupon period.
isin_code_s	char[12] A code that uniquely identifies an ISIN. The ISIN consists of: 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] Notation date in YYYYMMDD format.
deliverable_c	uint8_t Defines if a series can be delivered or not (cash settlement). Possible values include: 1 = Yes 2 = No.
effective_exp_date_s	char[8] The effective expiration date is the actual expiration date of the series and will normally be 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 will continue 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 ASX Trade with an external transaction. Possible values: 1= Yes 2 = No.



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



# 33 DQ128 Historical Combination Series

### 33.1 Query Function

This query is used to retrieve information about expired combination series. It is used by OI applications when viewing historical orders for expired combination series. Expired combination series are not available in response to a DQ126 Combination Series Query (refer to *DQ126 Combination Series*).

A combination (standard or tailor made) is always considered as unknown if the order history is from a previous date. A new combination created today may have the same binary code as a combination from a previous date, but the actual content may be different.

This accepts a list of unknown series. The response supplies information on each combination stored in the history matching the criteria given in the series list.

### 33.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_history_t	
Segmented	True	
Partitioned	False	
Answers	DA128	

### 33.3 Answer Properties

Transaction Type	DA128
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.

### 33.4 Related Messages

A related message includes query DQ127.

### 33.5 Message Structure

### 33.5.1 query_history_t

The query contains an item list where all the unknown series are sent.

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'D', 'Q', 128}.
series	series_t Set to zeros for information from all markets. Fill in country_c and market_c to retrieve information on that market. Fill in country_c, market_c and instrument_group_c to retrieve information on that instrument type.



Variable	Description
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
items_n	uint16_t The number of unknown combination series held in the item_t array.
date_trading_s	char[8] Not used by the query but a valid date must be supplied, e.g. today's date. Format: YYYYMMDD.
item	query_history_item_t[200] List of unknown combination series. See query_history_item_t sub structure below.

# 33.5.2 query_history_item_t

Variable	Description
series	series_t Identifies the unknown combination series.

### 33.6 Answer 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:

### • answer_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).

# 33.6.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'D', 'A', 128}.
items_n	uint16_t The number of top level items following this header.
size_n	uint16_t The size of the message, including this header.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2]



Variable	Description
	Ignore. Used for byte alignment.

# 33.6.2 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.

# 33.6.3 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.

# 33.6.4 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 expirations, the value in this field will be 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 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 in YYYYMMDD format. This date here, together with the time_first_trading_s field provides the first trading time in UTC.



Variable	Description
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	<ul> <li>uint8_t</li> <li>The actual state of the series. Possible values include:</li> <li>1 = Active (both expired and not expired)</li> <li>2 = Suspended (temporarily stopped - if an underlying is suspended then all derived series inherit that status)</li> <li>3 = Issued (series exists in the system but is unavailable for trading – to the user this is the same as delisted)</li> <li>4 = Delisted (indicates that the series has been permanently stopped from trading before its expiry).</li> </ul>
suspended_c	uint8_t Specifies whether the series is suspended or not. Possible values include: 1 = Yes 2 = No.
traded_in_click_c	uint8_t Indicates if this series is currently traded. Possible values include: 1 = Yes 2 = No.
db_operation_c	<ul> <li>uint8_t</li> <li>Operation that the trade system has carried out on this particular item.</li> <li><b>Note</b>: Users 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 user. As mentioned previously, a remove will be indicated by the named structure ns_remove_t (named structure 37002).</li> <li>Possible values:</li> <li>1 = Insert</li> <li>2 = Update.</li> </ul>
trade_reporting_only_c	uint8_t Ignore, not used. Always set to 2.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 33.6.5 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 include: B = Buy



Variable	Description
	S = Sell.
op_if_sell_c	char[1] Specifies whether to buy or sell the series when selling the combination. Possible values include: B = Buy S = Sell.



## 34 II17 Preliminary Settlement Prices

### 34.1 Query Function

This query makes it possible to retrieve information about preliminary settlement prices calculated by the exchange intraday. The information returned corresponds to the information that is broadcast in the Preliminary Settlement Price Update broadcast. Refer to *BI63 Settlement Price Update* in *ASX Trade Broadcasts* for more information.

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, and this query can be used to retrieve that same information. Users are required to enter the date of the settlement price they wish to retrieve.

### 34.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_prel_settlement_t	
Partitioned	False	
Segmented	True	
Answers	IA17	

### 34.3 Answer Properties

Transaction Type	IA17
Struct Name	answer_prel_settlement_t

## 34.4 Message Structure

## 34.4.1 query_prel_settlement_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'I', 'I', 17}.
series	series_t Set to zeros to return information on all instrument series where preliminary settlement prices have been calculated intraday. Fill in country_c, market_c and commodity_n to return only instrument series that match the given combination of fields.
settlement_date_s	char[8] Settlement date in YYYYMMDD format.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
settlement_price_type_c	uint8_t Different types of settlement prices. Possible values include:



Variable	Description
	1 = Apply all types
filler 1 s	char[1]
	Ignore. Used for byte alignment.

# 34.5 Answer Structure

A list of preliminary prices is returned to the sender.

# 34.5.1 answer_prel_settlement_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'I', 'A', 17}.
items_n	uint16_t The number of items held in the array.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
item	answer_prel_settlement_item_t[1500] See answer_prel_settlement_item_t sub structure below.

# 34.5.2 answer_prel_settlement_item_t

Variable	Description
series	series_t Contains the series to which this information relates.
settl_price_i	int32_t Settlement price.
settlement_date_s	char[8] Settlement date in YYYYMMDD format.
settlement_price_type_c	uint8_t The type of settlement price. Possible values include: 1 = Apply all types 2 = Normal.
hhmmss_s	char[6] Settlement time in HHMMSS format.
filler_1_s	char[1] Ignore. Used for byte alignment.



# 35 IQ12 Total Equilibrium Prices

## 35.1 Query Function

This transaction is used to download the equilibrium price information from ASX Trade.

The Equilibrium Price is only present in an overlapping market. It shows the price that the opening trades would trade at and the associated remaining quantity, if the series were to open.

To download data for all instrument series:

- 1. Remember the answer from the first query, the received instance_c.
- 2. Copy the value of the next_series field from the answer structure to the series field in the subsequent query.
- 3. Increment the value by one if the segment_number_n in the answer is greater than zero, and then copy it to the segment_number_n in the subsequent query.
- 4. If this is not the case, and the received sequence number is zero, the value of one should be copied.
- 5. Repeat steps 2 and 3 until instance_next_c in the answer is equal to the saved value from step 1, and the segment_number_n in the answer is zero.

### 35.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	query_tot_equil_prices_t
Partitioned	True
Segmented	True
Answers	IB12 (this is not an error - the response to IQ12 is IB12)

## 35.3 Answer Properties

Transaction Type	IB12
Struct Name	answer_tot_equil_prices_t

### 35.4 Message Structure

## 35.4.1 query_tot_equil_prices_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'I', 'Q', 12}.
series	series_t Fill in with any valid series.
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.



# 35.5 Answer Structure

# 35.5.1 answer_tot_equil_prices_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'I', 'B', 12}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
instance_c	uint8_t Indicates the Matching Engine partition from which this information has been retrieved.
next_instance_c	uint8_t Indicates the next Matching Engine partition that needs to be used for a subsequent query.
series_next	series_t Indicates which series identifier needs to be used for the subsequent query.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_tot_equil_prices_item_t[1230] See <i>answer_tot_equil_prices_item_t</i> sub structure below.

# 35.5.2 answer_tot_equil_prices_item_t

Variable	Description
series	series_t Contains the series to which this information relates.
equilibrium_quantity_i	int64_t The volume that will be executed at the equilibrium price once the instrument series goes into a trading state where order matching is enabled.
equilibrium_price_i	<ul> <li>int32_t</li> <li>The equilibrium price (also referred to as IAP - indicative auction price) 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.</li> <li>If the 31st bit (highest bit) is set and the rest are zero, then this indicates that there is no longer an equilibrium price for the series. Also equilibrium_quantity_i will be zero.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> <li>Note: If matching_price_type_c is 2, then this value indicates a reference price.</li> </ul>
best_bid_premium_i	int32_t 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. Decimal places are implied from the attributes of the associated instrument class to the series.
best ask premium i	int32 t



Variable	Description
	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. Decimal places are implied from the attributes of the associated instrument class to the series.
best_bid_quantity_i	int64_t The volume 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_bid_premium_i and the equilibrium_price_i values are equal then this quantity will be the surplus volume.
best_ask_quantity_i	int64_t The volume for the best ask price that will be in the order book when the instrument 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 quantity will be the surplus volume.
matching_price_type_c	uint8_t Different type of prices distributed as equilibrium price. Possible values include: 1 = Normal indicative Equilibrium Price 2 = Reference price. Not currently used
filler_3_s	char[3] Ignore. Used for byte alignment.



# 36 IQ18 Total Volumes and Query

### 36.1 Query Function

This query is used to download the Summary Price and order book Information from ASX Trade. In order to maintain a real time database of the information published, the OI application must listen to the BO14 and BD2 broadcasts.

The IQ18 provides the baseline information for the BO14 and BD2 broadcasts and the IQ19 query provides the baseline information for the BO15 and BD2. Both queries and their associated responses contain the same structures, however the information returned differs. IA18 will contain price depth, held in an array, whereas IA19 will return only the top of the market, held in the same array structure, but not using all the elements of that array.

Users should consider utilising this query only to match their subscribed broadcasts. In ASX Trade, the BO14 broadcast is only issued over derivative markets (Stock Derivatives market, Index Derivatives marketand Listed Funds, Structured Products & Warrants market); whereas the BO15 is issued over the Equity markets and the Interest Rate market. So although executing an IQ18 query over an Equity market is possible, the information returned can never be maintained through a subscription to a broadcast. Users should be tailoring their OI applications to use the IQ18 for instruments that have BO14 subscriptions and the IQ19 for instruments that have BO15 subscriptions.

- 1. The following sequence of actions must be performed by the application in order to synchronise this query's answer with the BO14 and BD2 broadcasts.
- 2. Start subscribing for BO14 and BD2 broadcasts.
- 3. Received broadcasts must **not** be processed until step 3. The user application must keep these broadcasts in an internal queue.
- 4. Send in the IQ18 transactions.
- 5. When the IQ18 download of data is complete, the user application must handle the queued BO14 and BD2 broadcasts.
- 6. Process the BO14 and BD2 broadcasts in the same order as they were received.
- 7. The application has the correct information at the point when all queued broadcasts have been handled.
- 8. Remove the usage of the internal queue when all queued broadcasts have been processed.
- 9. New broadcasts received should be used to directly modify the application's maintained database.

To download data for all authorised instrument series:

### 10. Copy the value in the next_series field to the series field of query_hdr_t in the subsequent query.

11. Increment the value by one if the segment_number_n in the answer is greater than zero.

12. Repeat steps 10 and 11 until segment_number_n and series_next in the answer is zero.

This answer will contain only series to which the user has been granted access. Novice OI developers can become confused when they receive price information for combinations which are unknown to them because they have yet to execute a DQ126 query to retrieve the existing combinations or subscribe for BU126 broadcasts for any new combinations. Also the answer will include each series regardless of whether it has any price information at that time.

The answer will contain only series to which the user has been granted access. Also, the answer will include each series regardless of whether it has any price information at that time.



#### Note:

When requesting IQ18 information without first having executed a DQ126 query to retrieve the existing combinations or subscribe for BU126 broadcasts for any new combinations, the IA18 may return information for unknown combination series.



## 36.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.
Partitioned	False
Segmented	True
Answers	IA18

### 36.3 Answer Properties

Transaction Type	IA18
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs are described below.

## 36.4 Message Structure

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

- query_hdr_t
- sub_item_hdr_t
- ob_levels_query_data_t (named structure 33020)
- optionally followed by:
  - sub_item_hdr_t
  - ob_levels_id_t (named structure 33002).

Users need to be aware that the ob_levels_query_data_t struct is mandatory as it contains the segment number field, whereas the ob_levels_id_t struct is optional as it contains the filtering series field. Users should not add or remove structures from subsequent queries once the initial query has been transmitted and its response received.

# 36.4.1 query_hdr_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'I' (O' 18}
series	series_t Filled with any valid series. This field indicates to ASX Trade where to start the search for price information. In subsequent queries this field should be filled with the values returned in the series_next field of the ob_levels_next_query_t struct in the answer.
items_n	uint16_t The number of items held in the VIM, either one or two depending if the user wishes to use the ob_levels_id_t struct.
size_n	uint16_t The size of the entire VIM, including this header.


## 36.4.2 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.

#### 36.4.3 ob_levels_query_data_t (named structure 33020)

Variable	Description
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 36.4.4 ob_levels_id_t (named structure 33002)

This named structure is not mandatory in the query. If the series is provided by the user however, then this acts as a filter for this query. Only instrument series matching the filter are returned in the answer. The structure may not be repeated within the query and each subsequent query used to retrieve further segments or partitions should contain the same filter information.

Variable	Description
series	series_t Acts as a filter for this query. Fill in country and market code to retrieve information on that particular market. Fill in country, market and instrument group code to retrieve information on that particular instrument type. Fill in country, market, instrument group and commodity code to retrieve information on that particular instrument class. Or provide a full series definition to query for a particular instrument series.
block_n	uint32_t This field is not used and should be set to zero.

### 36.5 Answer Structure

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

- answer_hdr_t
- sub_item_hdr_t
- ob_levels_next_query_t (named structure 33032)
- one or more sequences of:
  - sub_item_hdr_t
  - ob_levels_id_t (named structure 33002)
  - one or more sequences of:
    - sub_item_hdr_t
    - a choice of:
      - ob_levels_price_volumes_t (named structure 33003)



- ob_levels_order_number_t (named structure 33004)
- ob_levels_total_quantity_t (named structure 33005)
- ob_levels_no_of_orders_t (named structure 33033)
- market_info_base_t (named structure 33034)
- market_info_asx_t (named structure 33039)
- market_info_index_t (named structure 33040)
- ob levels undisclosed quantity t (named structure 33041)
- ob_levels_closing_t (named structure 33031)
- ob_levels_price_t (named structure 33006)
- market_info_asx_extended (named structure 33139)

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. The two of them are never distributed simultaneously. Currently, only the latter is distributed. The ob_levels_price_t struct is given here for future reference.

The interpretation of the various possible structures returned in the answer are the same as in BO14 with some additions and exceptions described in the structure definitions below.

### 36.5.1 answer_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'I', 'A', 18}.
items_n	uint16_t The number of sub items held in the VIM.
size_n	uint16_t The size of the entire message, including this header.

#### 36.5.2 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.

#### 36.5.3 ob_levels_next_query_t (named structure 33032)

This structure is used by the application in order to perform a complete download of information as previously described.

Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero indicates that there are no more segments.
instance_c	uint8_t Set to one.
instance_next_c	uint8_t Set to one.



Variable	Description
series_next	series_t
	Indicates the series identifier which needs to be used for the subsequent query.

#### 36.5.4 ob_levels_id_t (named structure 33002)

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

Variable	Description
series	series_t Contains the series to which this price information relates.
block_n	uint32_t This field is not used and should be set to zero.

### 36.5.5 ob_levels_price_volumes_t (named structure 33003)

The price masks are interpreted as bit fields where currently the lowest 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 will be 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. The two of them are never distributed simultaneously. Currently, only the ob_levels_price_volumes_t struct is distributed.

Variable	Description
bid_mask_n	uint16_t
	Interpreted as a bit field where the lowest 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 the broadcast. 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_c field holds the total number of items within the array.
	If the bid price mask has the value 3 (bit 0 and 1 set) and the ask price mask has the value 1 (bit 0 set only), the array will contain three items consisting of: Array[0] - Premium and demand for bid level 1 Array[1] - Premium and demand for bid level 2
	Array[2] - Premium and demand for ask level 1. This response indicates that there are not bid levels of 3, 4, and 5, or ask levels for 2, 3, 4 and 5.
	Example:
	If the bid price mask has the value of zero and the ask price mask has the value 31 (bit 0 to 4 set), the array consists of the following items: Array[0] - Premium and demand for ask level 1
	Array[1] - Premium and demand for ask level 2



Variable	Description
	Array[2] - Premium and demand for ask level 3 Array[3] - Premium and demand for ask level 4 Array[4] - Premium and demand for ask level 5. This response indicates that there are no orders at all on the bid side for this series.
ask_mask_n	uint16_t Refer to the description for bid_mask_n above.
premium_levels_c	<ul> <li>uint8_t</li> <li>Propagates the currently distributed price depth for this instrument series. Possible values are currently in the range from zero to five. This value can be changed up to a maximum of 10.</li> <li>0 = the exchange doesn't distribute any prices at all</li> <li>1 = the exchange distributes the first ranked price level</li> <li>2 = the exchange distributes two best prices levels</li> <li>3 = the exchange distributes four best prices levels</li> <li>4 = the exchange distributes five best prices levels</li> <li>5 = the exchange distributes five best prices levels.</li> <li>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.</li> </ul>
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 1 (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 below.

# 36.5.6 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).

## 36.5.7 ob_levels_order_number_t (named structure 33004)

The order numbers provided in this structure are the order numbers for the first ranked bid and ask orders in the order book.



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.

### 36.5.8 ob_levels_total_quantity_t (named structure 33005)

The quantities provided in this structure are the total quantity of all orders in the order book for the instrument series.

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).

## 36.5.9 ob_levels_no_of_orders_t (named structure 33033)

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).
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 Number of levels in aggregated price depth.
filler_2_s	char[2] Ignore. Used for byte alignment.
items_c	uint8_t The number of items held in the array.
item	ob_levels_no_of_orders_item_t[32] See ob_levels_no_of_orders_item_t sub structure below.

## 36.5.10 ob_levels_no_of_orders_item_t

Variable	Description
no_of_orders_u	uint32_t Number of orders for this price level.

## 36.5.11 market_info_base_t (named structure 33034)

This structure is provided in the answer only if any of the price related fields have a value.



Variable	Description
opening_price_i	int32_t The first on-market sale price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
high_price_i	<ul> <li>int32_t</li> <li>The highest on-market traded price of the series for the day.</li> <li>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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
low_price_i	int32_t The lowest on-market traded price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
last_price_i	int32_t The last on-market traded price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
volume_u	int64_t Volume of the last trade.
turnover_u	int64_t Total traded volume during the day.
number_of_deals_u	uint32_t Number of trades executed.
hhmmss_s	char[6] The UTC (local time) when this processed price information was issued. This will be the same value as shown in the last BD2 broadcast given for this particular series. Given in HHMMSS format.
trend_indicator_c	char[1] Trend indicator for new price compared to previous one: '+' = higher price than previously '-' = lower price than previously '=' = same price as previously ' ' = no trend available.
	unito_t



Variable	Description
	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.

## 36.5.12 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.

## 36.5.13 market_info_index_t (named structure 33040)

This structure is provided in the answer only if any of the price related fields have a value.

Variable	Description
high_price_i	<ul> <li>int32_t</li> <li>The highest value for an index during the day.</li> <li>If the 31st 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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
low_price_i	<ul> <li>int32_t</li> <li>The lowest value during the day.</li> <li>If the 31st 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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
last_price_i	<ul> <li>int32_t</li> <li>The last value given during the day.</li> <li>If the 31st 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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
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



Variable	Description
	Points change since the current day's initial value. Decimal places are implied from the attributes of the associated instrument class to the series.
date_time_of_dist_s	char[14] UTC and time of distribution in YYYYMMDDHHMMSS format.
date_time_of_comp_s	char[14] UTC date and time of computation in YYYYMMDDHHMMSS format.

## 36.5.14 ob_levels_undisclosed_quantity_t (named structure 33041)

This structure will show whether a price level, each side of the book, has Undisclosed Quantity or not. 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).	

### 36.5.15 ob_levels_closing_t (named structure 33031)

This structure is provided in the answer only if any of the included fields has a value set.

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).

### 36.5.16 ob_levels_price_t (named structure 33006)

This structure is used in the same way as ob_levels_price_volumes_t.

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. The two of them are never distributed simultaneously. Currently, only the latter is distributed. The ob_levels_price_t struct is given here for future reference.

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).



Variable	Description
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).
premium_levels_c	uint8_t Number of levels in aggregated price depth.
demands_populated_c	uint8_t Indicates if the distribution of quantity is enabled or disabled for the different price levels. Possible values: 1 = Demands are populated 2 = Demands are not populated.
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_item_t[32] See ob_levels_price_item_t sub structure below.

## 36.5.17 ob_levels_price_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 premium 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.

## 36.5.18 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.
extended_high_price_i	int64_t This field may be set to the high price with up to four decimal places.
extended_low_price_i	int64_t This field may be set to the low price with up to four decimal places.
extended_last_price_i	int64_t This field may be set to the low price with up to four decimal places.

## 36.6 Examples

The examples below illustrate the functionality of IA18/IA19 with respect to what information they may contain in different market situations.



#### 36.6.1 Example 1

Before opening on a derivatives market, no orders have been entered and no price or volume statistics are available. The reply will consist only of the structures containing series information. Then for each series in the answer, a sequence number structure for synchronising the query and broadcast flow and a closing price structure is sent. As with all responses to this query, the reply will contain information on what to send in the next query. This may include:

- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_closing_t
- ob levels id t
- ob_levels_closing_t
- ob_levels_id_t (no closing price or open balance available)
- ob levels id t (no closing price or open balance available)
- ob_levels_id_t (no closing price or open balance available)
- ob_levels_closing_t.

#### 36.6.2 Example 2

When the query is placed after the market has opened and there are orders in the market, and trades have been matched, then the sequence of named structures may include:

- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- market_info_base_t
- ob_levels_closing_t
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- market_info_base_t
- ob_levels_closing_t
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t.

#### 36.6.3 Example 3

When the query is placed after the market has opened and there are orders in the market but no trades have been matched, the sequence of named structures may include:

- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- ob_levels_closing_t (closing price or open balance set)
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t
- ob_levels_id_t
- ob levels price volumes t (or ob levels price t)
- ob_levels_closing_t (closing price or open balance set)



- ob_levels_order_number_t (if enabled)
  ob_levels_total_quantity_t (if enabled)
  ob_levels_no_of_orders_t.



## 37 IQ19 Total Volumes and Prices Query

#### 37.1 Query Function

This query is used to download the Summary Price and order book Information from ASX Trade. In order to maintain a real time database of the information published, the OI application must listen to the BO15 and BD2 broadcasts.

The IQ19 provides the baseline information for the BO15 and BD2 broadcasts, and the IQ18 broadcasts provide the baseline information for the BO14 and BD2 broadcasts. Both queries and their associated responses contain the same structures, however the information returned differs. IA18 will contain price depth, held in an array, whereas IA19 will return only the top of the market, held in the same array structure, but not using all the elements of that array.

Users should consider utilising this query only to match their subscribed broadcasts. In ASX Trade, the BO14 broadcast is only issued over derivative markets (Stock Derivatives market, Index Derivatives market, and Listed Funds, Structured Products and Warrants market), whereas the BO15 is issued over the Equity market and the Interest Rate market). So although executing an IQ18 query over an Equity market is possible, the information returned can never be maintained through a subscription to a broadcast. Users should be tailoring their OI applications to use the IQ18 for instruments that have BO14 subscriptions.

To synchronise the query, answer with BO15 and BD2 broadcasts:

- 1. Start subscribing for BO15 and BD2 broadcasts.
- 2. Received broadcasts must not be processed until step 3. The user application must keep these broadcasts in an internal queue.
- 3. Send in the IQ19 transaction.
- 4. When the IQ19 download of data is complete, the user's application must handle the queued BO15 and BD2 broadcasts.
- 5. Process the BO15 and BD2 broadcasts in the same order as they were received.
- 6. The application has the correct information at the point when all queued broadcasts have been handled.
- 7. Remove the usage of the internal queue.
- 8. New broadcasts received should directly modify the application's maintained database.

To download data for all authorised instrument series:

- 9. Remember the answer for the first received query instance_c.
- 10. Copy the value in the next_series field to the series field of query_hdr_t in the subsequent query.
- 11. Increment the value by one if the segment_number_n in the answer is greater than zero.
- 12. Repeat steps 10 and 11 until segment_number_n and series_next in the answer is zero.

This answer will contain only those series to which the user has been granted access. Also the answer will include each series regardless of whether it has any price information at that time.



#### Note:

When requesting IQ19 information without first having executed a DQ126 query to retrieve the existing combinations or subscribe for BU126 broadcasts for any new combinations, the IA19 may return information for unknown combination series.

#### 37.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO



Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.
Partitioned	False
Segmented	True
Answers	IA19

#### 37.3 Answer Properties

Transaction Type	IA19
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs is described below.

### 37.4 Message Structure

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

- query_hdr_t
- sub_item_hdr_t
- ob_levels_query_data_t (named structure 33020)
- optionally followed by:
  - sub_item_hdr_t
    - ob_levels_id_t (named structure 33002).

Users need to be aware that the ob_levels_query_data_t struct is mandatory as it contains the segment number field, where as the ob_levels_id_t struct is optional as it contains the filtering series field. Users should not add or remove structures from subsequent queries once the initial query has been transmitted and its response received.

#### 37.4.1 query_hdr_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'I', 'Q', 19}.
series	series_t Filled with any valid series. This field effectively tells ASX Trade where to start the search for price information. In subsequent queries this field should be filled with the values returned in the series_next field of the ob_levels_next_query_t struct in the answer.
items_n	uint16_t The number of items held in the VIM, either one or two depending if the user wishes to use the ob_levels_id_t struct.
size_n	uint16_t The size of the entire VIM, including this header.

### 37.4.2 sub_item_hdr_t

Variable	Description
named_struct_n	uint16_t Contains a number, indicating the type of structure that follows.



Variable	Description
size_n	uint16_t
	The size of the structure that is to follow, including this header.

### 37.4.3 ob_levels_query_data_t (named structure 33020)

Variable	Description
segment_number_n	uint16_t Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 37.4.4 ob_levels_id_t (named structure 33002)

This named structure is not mandatory in the query. If provided by the user however, then the series is used as a filter for this query. Only instrument series matching the filter is returned in the answer. The structure may not be repeated within the query and each subsequent query should contain the same series information.

Variable	Description
series	series_t Acts as a filter for this query. Fill in country and market code to retrieve information on that particular market. Fill in country, market and instrument group code to retrieve information on that particular instrument type. Fill in country, market, instrument group and commodity code to retrieve information on that particular instrument class. Or provide a full series definition to query for a particular instrument series.
block_n	uint32_t This field is not used and should be set to zero.

## 37.5 Answer Structure

- answer_hdr_t
- sub_item_hdr_t
- ob_levels_next_query_t (named structure 33032)
- one or more sequences of:
  - sub_item_hdr_t
  - ob_levels_id_t (named structure 33002)
  - one or more sequences of:

### sub_item_hdr_t

- a choice of:
  - ob_levels_price_volumes_t (named structure 33003)
  - ob_levels_order_number_t (named structure 33004)
  - ob_levels_total_quantity_t (named structure 33005)
  - ob_levels_no_of_orders_t (named structure 33033)
  - ob_levels_price_t (named structure 33006)
  - market_info_base_t (named structure 33034)
  - market_info_asx_t (named structure 33039)
  - market_info_index_t (named structure 33040)



- ob_levels_undisclosed_quantity_t (named structure 33041)
- ob_levels_closing_t (named structure 33031)
- market_info_asx_extended (named structure 33139)

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. The two of them are never distributed simultaneously. Currently, only the latter is distributed. The ob_levels_price_t struct is given here for future reference.

The interpretation of the various possible structures returned in the answer are the same as in BO15 with some additions and exceptions described in the structure definitions below.

### 37.5.1 answer_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'I', 'A', 19}.
items_n	uint16_t The number of sub items held in the VIM.
size_n	uint16_t The size of the entire message, including this header.

## 37.5.2 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.

### 37.5.3 ob_levels_next_query_t (named structure 33032)

This structure is used by the application in order to perform a complete download of information as previously described.

Variable	Description
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
instance_c	uint8_t Set to one
instance_next_c	uint8_t Set to one.
series_next	series_t Indicates the series identifier which needs to be used for the subsequent query.

### 37.5.4 ob_levels_id_t (named structure 33002)

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



Variable	Description
series	series_t Contains the series to which this price information relates.
block_n	uint32_t This field is not used and should be set to zero.

### 37.5.5 ob_levels_price_volumes_t (named structure 33003)

The price masks are interpreted as bit fields where currently the lowest 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 rank 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 the first level of depth will be disseminated. However users should be aware that the system is capable of 10 levels of depth in this response. The number of levels will be 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. The two of them are never distributed simultaneously. Currently, only the ob_levels_price_volumes_t struct is distributed.

Variable	Description
bid_mask_n	uint16_t
	Interpreted as a bit field where the lowest 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 the broadcast. 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 rank prices are placed before lower ranked prices in the array. The items_c field
	holds the total number of items within the array.
	Example:
	If the bid price mask has the value three (bit 0 and 1 set) and the ask price mask has the
	value one (bit 0 set only), the array will contain three items consisting of:
	Array[0] - Premium and demand for bid level 1
	Array[1] - Premium and demand for bid level 2
	Array[2] - Premium and demand for ask level 1.
	This response indicates that there are not bid levels of 3, 4, and 5, nor ask levels for 2, 3, 4
	and 5.
	Example:
	If the bid price mask has the value zero and the ask price mask has the value 31 (bit 0 to 4
	set), the array consists of the following items:
	Array[0] - Premium and demand for ask level 1
	Array[1] - Premium and demand for ask level 2
	Array[2] - Premium and demand for ask level 3
	Array[3] : Premium and demand for ask level 4
	Array[4] - Premium and demand for ask level 5.
	This response indicates that there are no orders at all on the bid side for this series.
ask_mask_n	uint16_t
	Refer to the description for bid_mask_n above.
premium levels c	uint8 t



Variable	Description
	<ul> <li>Propagates the currently distributed price depth for this instrument series. Possible values are currently in the range from zero to one. Note that this value can be changed up to a maximum of 10.</li> <li>0 = The exchange doesn't distribute any prices at all</li> <li>1 = That the exchange distributes the first ranked price level.</li> <li>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.</li> </ul>
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 will always be 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 below.

## 37.5.6 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 premium 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).

## 37.5.7 ob_levels_order_number_t (named structure 33004)

The order numbers provided in this structure are the order numbers for the first ranked bid and ask orders in the order book.

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.

## 37.5.8 ob_levels_total_quantity_t (named structure 33005)

The quantities provided in this structure are the total quantity of all orders in the order book for the instrument series.



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).

### 37.5.9 ob_levels_price_t (named structure 33006)

This structure is used in the same way as ob_levels_price_volumes_t.

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. The two of them are never distributed simultaneously. Currently, only the latter is distributed. The ob_levels_price_t struct is given here for future reference.

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).
premium_levels_c	uint8_t Number of levels in aggregated order depth.
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 will always be two (disabled) as the volume will not 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_item_t[32] See ob_levels_price_item_t sub structure below.

## 37.5.10 ob_levels_price_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 premium 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.



## 37.5.11 ob_levels_no_of_orders_t (named structure 33033)

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).
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 Number of levels in aggregated price depth.
filler_2_s	char[2] Ignore. Used for byte alignment.
items_c	uint8_t The number of items held in the array.
item	ob_levels_no_of_orders_item_t[32] See ob_levels_no_of_orders_item_t sub structure below.

## 37.5.12 ob_levels_no_of_orders_item_t

Variable	Description
no_of_orders_u	uint32_t
	Number of orders for this price level.

## 37.5.13 market_info_base_t (named structure 33034)

This structure is provided in the answer only if any of the price related fields have a value.

Variable	Description
opening_price_i	<ul> <li>int32_t</li> <li>The first on-market sale price of the series for the day.</li> <li>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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
high_price_i	int32_t The highest on-market traded price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
low_price_i	int32_t



Variable	Description
	The lowest on-market traded price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
last_price_i	int32_t The last on-market traded price of the series for the day. If the 31 st 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. Decimal places are implied from the attributes of the associated instrument class to the series.
volume_u	int64_t Volume of the latest trade.
turnover_u	int64_t Total traded volume during the day.
number_of_deals_u	uint32_t Number of trades executed.
hhmmss_s	char[6] The UTC when this processed price information was issued. This will be the same value as shown in the last BD2 broadcast given for this particular series. Given in HHMMSS format.
trend_indicator_c	char[1] Trend indicator for new price compared to previous one: '+' = higher price than previously '-' = lower price than previously '=' = same price as previously ' ' = no trend available.
deal_source_c	uint8_t Where the last trade was executed. Refer to <i>Trade Source</i> in <i>ASX Trade Information and</i> <i>Business Information</i> for the possible values in this field.

## 37.5.14 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.

# 37.5.15 market_info_index_t (named structure 33040)

This structure is provided in the answer only if any of the price related fields have a value.

Variable	Description
high_price_i	int32_t



Variable	Description
	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	<ul> <li>int32_t</li> <li>The lowest value for an index during the day.</li> <li>If the 31st 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.</li> <li>Decimal places are implied from the attributes of the associated instrument class to the series.</li> </ul>
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.
date_time_of_dist_s	char[14] UTC and time of distribution in YYYYMMDDHHMMSS format.
date_time_of_comp_s	char[14] UTC and time of computation in YYYYMMDDHHMMSS format.

## 37.5.16 ob_levels_undisclosed_quantity_t (named structure 33041)

This structure will show whether a price level, each side of the book, has Undisclosed Quantity or not. 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).



#### 37.5.17 ob_levels_closing_t (named structure 33031)

This structure is provided in the answer only if any of the included fields has a value set.

Variable	Description
closing_price_i	int32_t
	This field contains the previous day's closing price.
open_balance_u	int64_t Indicates the open interest on the series, i.e. the number of outstanding contracts (this is not updated intraday).

#### 37.5.18 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.
extended_high_price_i	int64_t This field may be set to the high price with up to four decimal places.
extended_low_price_i	int64_t This field may be set to the low price with up to four decimal places.
extended_last_price_i	int64_t This field may be set to the low price with up to four decimal places.

#### 37.6 Examples

The examples below illustrate the functionality of IA18/IA19 with respect to what information they may contain in different market situations.

#### 37.6.1 Example 1

Before opening on a derivatives market no orders have been entered, and no price or volume statistics are available. The reply will consist only of the structures containing series information. Then for each series in the answer, a sequence number structure for synchronising the query and broadcast flow and a closing price structure is sent. As with all responses to this query, the reply will contain information on what to send in the next query. The sequence of named structures may look like:

- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_closing_t
- ob_levels_id_t
- ob_levels_closing_t
- ob_levels_id_t (no closing price or open balance available)
- ob_levels_id_t (no closing price or open balance available)
- ob_levels_id_t (no closing price or open balance available)
- ob_levels_closing_t.

#### 37.6.2 Example 2

When the query is placed after the market has opened and there are orders in the market, and trades have been matched, then the sequence of named structures may look like:



- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- market_info_base_t
- ob_levels_closing_t
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- market_info_base_t
- ob_levels_closing_t
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t.

#### 37.6.3 Example 3

When the query is placed after the market has opened and there are orders in the market but no trades have been matched, the sequence of named structures may look like:

- ob_levels_next_query_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- ob_levels_closing_t (closing price or open balance set)
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t
- ob_levels_id_t
- ob_levels_price_volumes_t (or ob_levels_price_t)
- ob_levels_closing_t (closing price or open balance set)
- ob_levels_order_number_t (if enabled)
- ob_levels_total_quantity_t (if enabled)
- ob_levels_no_of_orders_t.



## 38 IQ48 Settlement Prices Query

#### 38.1 Query Function

This query is used to retrieve settlement prices for either today or the previous day.

The information is available sometime after the market has closed. The BI7 broadcast indicates that the information is ready (refer to *BI7 Information Ready* in *ASX Trade Broadcasts* for more information).

If a particular series does not have a settlement price for the relevant date, then it will not appear in the response.

### 38.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP4	
Struct Name	query_trade_statistics_settle_t	
Partitioned	False	
Segmented	True	
Answers	IA48	

## 38.3 Answer Properties

Transaction Type	IA48
Struct Name	answer_trade_statistics_settle_t

#### 38.4 Message Structure

#### 38.4.1 query_trade_statistics_settle_t

Variable	Description
transaction_type	transaction_type_t
	Set the structure to the following: {'l', 'Q', 48}.
series	series_t
	Fill in the country_c and market_c fields to receive information on that particular market.
segment_number_n	uint16_t
	Indicates which segment users wish to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
date_s	char[8]
	Indicates which date the user wishes to receive, either today or the previous business day.
	Format: YYYYMMDD.
filler_2_s	char[2]
	Ignore. Used for byte alignment.

### 38.4.2 Return Codes

An IQ48 query may be aborted by ASX Trade, in which case only the reason for the query being aborted is returned to the sender.



Cstatus	Txstat
Successful	INFO_SUCCESS – successful completion.
Successful	INFO_NOINFO - no information available for specified segment, market and date.
Successful	INFO_TODAYNOTAVAIL – today's information is not yet available.
Transaction aborted	INFO_BADSEG – segment number cannot be zero in an input query.

## 38.5 Answer Structure

## 38.5.1 answer_trade_statistics_settle_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'I', 'A', 48}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_trade_statistics_settle_item_t[3500] See answer_trade_statistics_settle_item_t sub structure below.

## 38.5.2 answer_trade_statistics_settle_item_t

Variable	Description
series	series_t Contains the series to which this price information relates.
settle_price_i	int32_t The daily settlement price for the series.



## 39 ML1 Price Limit Query

This query returns the Price Limit Reference Price and Price Limits for instrument series.

## 39.1 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_limit_trans_t	
Partitioned	True	
Segmented	False	
Answers	MA1	

### 39.2 Answer Properties

Transaction Type	MA1
Struct Name	answer_limit_t

## 39.3 Message Structure

## 39.3.1 query_limit_trans_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'L', 1}.
series	series_t To retrieve all price limits, on the first query zero-fill the series. On subsequent queries users are to fill this field with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field can also act as a filter for a particular instrument series. Fill in the entire structure to indicate that the response should only contain the price limits for that series. The user must set the only_this_series_c field to 1 in this case. The final response is indicated by the series field in the answer being zero-filled.
only_this_series_c	uint8_t Only orders for a specific series are requested. Possible values include: 0 = Price Limits on all series 1 = Price Limits on the particular series indicated in the series field.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 39.4 Answer Structure

## 39.4.1 answer_limit_t

Variable	Description
transaction_type	transaction_type_t



Variable	Description
	Contains the following: {'M', 'A', 1}.
series	series_t The value in this field should be used in the next query. A zero-filled structure indicates this is the last answer.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_limit_item_t[1000] See answer_limit_item_t sub structure below.

# 39.4.2 answer_limit_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.



#### 40 MQ7 Total Order Book Query

This query returns all orders in the order book.

#### 40.1 Query Function

#### 40.1.1 Snapshot Issue

When the response for any one series in an MQ7 query spans multiple segments, ASX Trade will cache the response for that series at the time the MQ7 was received so that the response will contain a true snapshot of the series. This means that all segments of the response for that series will have the same sequence number. The sequence number corresponds to the associated BO2 broadcasts (refer to *BO2 Order Book Changes* in *ASX Trade Broadcasts* for more information). This applies whether the original query is for a single series or is a wildcard request.

When a query for a series is started, if the series has more than 1000 orders (or 400 in a market where participant IDs are disclosed such as the Warrants market), the order book process will delete any current snapshot for that user, then take a complete snapshot of the order book for that series. It will then return the first segment. The user then requests the second segment using the details returned in the first response. The system checks its cached snapshot for that user, and if the series match, it will return the orders from the cached data.

The MQ7 may be rejected with an ME_OHS_DATAINCOMPLETEerror code (-485014). This should be treated as a temporary error and the query for that partition restarted (by setting all fields other than commodity_n to zero, which should be set to the commodity code of the first underlying in that partition which failed to complete its download).

This error will occur:

- If a segment is requested and the backend cannot find the snapshot of the instrument.
- If the initial query is done on one session and the second segment requested on a different user session.
- If the initial query is done on one session and all of the segment requests for the same partition are not completed within 60 seconds.
- If the initial query for one series is done, then an initial query for a second series, then the second segment for the first series is done.
- If the initial query is done and then the second segment requested and the system has failed over to the standby processor between the requests.

Furthermore, users are advised that in the case where a query for a series is started and between the segments the system fails over to the backup processor, if the next segment requested is the first on the sell side, the query will be responded to normally but the snapshot on the sell side may not match the snapshot of the buy side. Despite this, the integrity will be correct within each side of the market. If the failover to the standby system occurs other than on the switch between buy and sell orders, an error will be returned as outlined above.

#### 40.1.2 Recommended Practice

With this extra error condition and as per *Conducting a Standard Session* in *ASX Trade Introduction and Business Information*, ASX recommends the following practice of processing MQ7 requests.

For individual series:

- Subscribe for BO2 broadcasts for specific series (if maintaining a dynamic Order book Equity and Interest Rate markets only) and queue them for subsequent processing.
- Submit MQ7 query for specific series.
- Restart the query if an error is received.
- After successfully receiving all segments of the query, process any queued broadcasts by discarding those BO2 broadcasts for this series that have a lower or the same sequence number.



• Once all queued broadcasts are processed, a current up to date market picture has been achieved. For wildcard queries:

- Subscribe for BO2 broadcasts (if maintaining a dynamic order book Equity and Interest Rate markets only) and queue them for subsequent processing.
- Submit MQ7 query for wildcard and store BO2 sequence number for each series.
- If an error is received, restart query at beginning of current series (set series to current series, bid_or_ask_c to 0, order number to 0, this_series_only_c to false).
- After successfully receiving all segments of the query, process any queued broadcasts by discarding those BO2 broadcasts for each series that have a lower or the same sequence number.
- Once all queued broadcasts are processed, a current up to date market picture has been achieved.

#### 40.1.3 No Depth Issue

If there happens to be no orders in a series, the response to the MQ7 will not retrieve any data. Therefore there will be no sequence number corresponding to this particular series. If a broadcast for that series is received, it should be applied. In this case, users will be required to ignore any processing errors that occur at this point (i.e. processing a BO2 that updates a non-existent order, or processing a BO2 that deletes a non-existent order). Doing so will achieve a correct market picture.

However, the issue lies in when to stop ignoring these errors from processing BO2s for this particular series. ASX currently recommends that users:

- Record the point in the broadcast stream at which the MQ7 query has finished.
- Execute one more broadcast poll to ensure that all BO2 broadcasts have been retrieved at the time.
- Apply the BO2 broadcasts up to this point ignoring any errors.
- Apply subsequent broadcasts normally.

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	 query_tot_ob_t	
Partitioned	True	
Segmented	False	
Answers	MA42	

#### 40.2 Query Properties

### 40.3 Answer Properties

Transaction Type	MA42
Struct Name	answer_tot_ob_t

### 40.4 Message Structure

#### 40.4.1 query_tot_ob_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 7}.



Variable	Description
series	series_t To retrieve all orders in the system on the first query zero-fill the series, order_number_u and bid_or_ask_c fields. On subsequent queries users are to fill in the same three fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field can also act as a filter for a particular instrument series. Fill in the entire structure to indicate that the response should only contain the orders for that series. The user must set the only_this_series_c field to 1 in this case. The final response is indicated by the series field in the answer being zero-filled.
order_number_u	quad_word Used as a coupling field for retrieving partitioned information. Use the series, this and the bid_or_ask_c field to retrieve partitioned information.
bid_or_ask_c	uint8_t Used as a coupling field for retrieving partitioned information. Use the series, order_number_u and this field to retrieve partitioned information.
only_this_series_c	uint8_t Only orders for a specific series are requested. Possible values include: 0 = Orders on all series 1 = Orders on the particular series indicated in the series field.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 40.5 Answer Structure

## 40.5.1 answer_tot_ob_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 42}.
series	series_t The value in this field, along with the order_number_u and bid_or_ask_c values, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_number_u	quad_word The value in this field, along with the series and bid_or_ask_c values, should be used in the next query.
items_n	uint16_t The number of items held in the array.
bid_or_ask_c	uint8_t The value in this field, along with the order_number_u and series values, should be used in the next query.
filler_1_s	char[1] Ignore. Used for byte alignment.
item	answer_tot_ob_item_t[1000] See <i>answer_tot_ob_item_t</i> sub structure below.



## 40.5.2 answer_tot_ob_item_t

Variable	Description
order_number_u	quad_word The order identifier for the order in this item.
sequence_number_u	uint32_t Sequence number for the order in this item. Can be used to synchronise this query with the BO2 Order broadcast.
ob_position_u	uint32_t Indicates the order book position, 1 being the highest priority.
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
filler_3_s	char[3] Ignore. Used for byte alignment.
order_no_id	order_no_id_t See order_no_id_t sub structure below.
party	party_t See party_t sub structure below.

## 40.5.3 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.
	Note that the price for a combination 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, Centre Point or Sweep will not be shown in this response.
bid_or_ask_c	uint8_t
	Bid or Ask. Possible values include:
	1 = Buy
	2 = Sell.
filler_1_s	char[1]
	Ignore. Used for byte alignment.



## 40.5.4 party_t

This structure will be blanked for instrument series that do not disclose participant IDs.

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.

## 40.5.5 Return Codes

As mentioned in the preamble for this query, it is possible to receive an error if the snapshot is not available. Users should re-query at the current series.

Cstatus	Description
Successful	Snapshot of order depth was returned
Transaction aborted	ME_OHS_DATAINCOMPLETE – Snapshot could not be returned



## 41 MQ8 Total Order Query

## 41.1 Query Function

Note:

This is used to query the user's own orders in the order book.

# (j)

This query is a legacy query that does not return all order details. It only returns those fields that can be specified for an order using the legacy MO31 Enter Order transaction. To retrieve the full order details, use the MQ92 Own Orders query.

## 41.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_tot_order_t	
Partitioned	true	
Segmented	false	
Answers	MA43	

## 41.3 Answer Properties

Transaction Type	MA43
Struct Name	answer_tot_order_t

#### 41.4 Message Structure

### 41.4.1 query_tot_order_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 8}.
series	series_t To retrieve a participants own orders in the system on the first query zero-fill this series field and the order_index_u field. On subsequent queries users are to fill in the same two fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field cannot act as a filter. The query retrieves all orders for a particular participant or user.
whose	whose_t Used to filter the results. See whose_t sub structure below.
order_index_u	uint32_t Used as a coupling field for retrieving partitioned information. Use the series and this field to retrieve partitioned information.



## 41.4.2 whose_t

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

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



Note:

Fields that are omitted should be filled with NULLs and are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] When placing an order, this field is a free text and is typically used to indicate to participants the ultimate client making the order. In this structure it can be used as a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 41.4.3 trading_code_t

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

### 41.5 Answer Structure

After a successful MQ8 transaction, a list of own orders in the order book is returned to the sender.



## 41.5.1 answer_tot_order_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 43}.
series	series_t The value in this field, along with the order_index_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_index_u	uint32_t The value in this field, along with the series value, should be used in the next query.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_tot_order_item_t[300] See answer_tot_order_item_t sub structure below.

## 41.5.2 answer_tot_order_item_t

Variable	Description
order_number_u	quad_word The order identifier for the order in this item.
sequence_number_u	uint32_t Ignore. Currently not used.
ob_position_u	uint32_t Indicates the order book position, 1 being the highest priority.
combo_mark_c	uint8_t Ignore. Not used in this query.
filler_3_s	char[3] Ignore. Used for byte alignment.
party	party_t See party_t sub structure.
order	order_t Detailed information on the order. See order_t sub structure.
total_volume_i	<pre>int64_t The total quantity of the order, i.e. both the hidden and shown (display_quantity_i) volume. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
display_quantity_i	int64_t This value should only be considered if the order is an iceberg order. It indicates the original displayed quantity of the order and is therefore the value which the iceberg order



Variable	Description
	will use to re-populate its display_quantity_i field once all its current shown quantity has been traded. ASX Trade will automatically populate this field when the total_volume_i field != 0.
orig_shown_quantity_i	int64_t The shown quantity of the order when it was originally entered.
orig_total_volume_i	int64_t The total_volume_i value of the order when it was originally entered.
timestamp_in	time_spec_t The time when the order was entered <b>or</b> when it was last modified. See time_spec_t sub structure.
timestamp_created	time_spec_t The time when the order was originally entered. See time_spec_t sub structure.

## 41.5.3 party_t

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

# 41.5.4 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 for more information.
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]


Variable	Description
	A free text field used at 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. 31 indicates
	MO31).
change_reason_c	uint8_t
	Indicates the reason for the last order 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 trade system
	10 = Order deleted by proxy
	13 = Hidden volume order recalculated
	14 = Covered option delete
	19 = Trade system deleted day order
	20 = Deleted by system due to Instrument Session change
	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 EQ = Success order releaded without MAQ and mid tick attribute
	SU = Sweep order reloaded without MAQ and mid-tick attribute.
filler_1_s	char[1]
	Ignore. Used for byte alignment.

## 41.5.5 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.
premium_i	int32_t



Variable	Description
	The price of the order. A combination of this field and the order_type_c field signify different types of orders. 0 = Market order (order_type_c > 1) Any value = Fixed price order (order_type_c=1). Note that the price for a tailor made combination order can be positive, zero or negative.
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	<ul> <li>uint16_t</li> <li>This field is made up of two 8 bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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)</pre>
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.



Variable	Description
	sher [15]
customer_info_s	Char[15]
	customer information – a nee text field typically used to indicate to the participant their
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 codes. Possible values:
	0 = Ignore, not relevant.
	For other values refer to Trade Report Types in ASX Trade Introduction and Business
	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 include:
	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.
stop condition c	uint8_t
	Ignore. Currently not used.
filler_2_s	char[2]
	Ignore. Used for byte alignment.

# 41.5.6 trading_code_t/ex_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 three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.



## 41.5.7 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 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.

# 41.5.8 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.



### 42 MQ9 Total Inactive Order Query

#### 42.1 Query Function

This query retrieves the participants own inactive orders in the order book. The query and answer use the same structures as the MQ8/MA43, only the values for the transaction_type_t differ.



## Note:

This query is a legacy query that does not return all order details. It only returns those fields that can be specified for an order using the legacy MO31 and MO98 transactions. To retrieve the full order details, use the MQ92 Own Orders query, which can also be used to query for inactive orders.

### 42.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_tot_order_t	
Partitioned	True	
Segmented	False	
Answers	MA44	

#### 42.3 Answer Properties

Transaction Type	MA44
Struct Name	answer_tot_order_t

### 42.4 Message Structure

#### 42.4.1 query_tot_order_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 9}.
series	series_t To retrieve all own inactive orders in the system on the first query zero-fill this series field and the order_index_u field. On subsequent queries users are to fill in the same two fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field cannot act as a filter. The query retrieves all own inactive orders.
whose	whose_t This can be used to filter the results. See whose_t sub structure below.
order_index_u	uint32_t Used as a coupling field for retrieving partitioned information. Use the series and this field to retrieve partitioned information.



## 42.4.2 whose_t

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

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



### Note:

Fields that are omitted should be filled with NULLs and are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] When placing an order, this field is a free text field and is typically used to indicate to ASX participants the ultimate client making the order. In this structure it can be used as a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 42.4.3 trading_code_t

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



## 42.5 Answer Structure

## 42.5.1 answer_tot_order_t

After a successful MQ9 transaction, a list of own inactive orders in the order book is returned to the sender.

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 44}.
series	series_t The value in this field, along with the order_index_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_index_u	uint32_t The value in this field, along with the series value, should be used in the next query.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_tot_order_item_t[300] An array containing the order details. See answer_tot_order_item_t sub structure below.

## 42.5.2 answer_tot_order_item_t

Variable	Description
order_number_u	quad_word
	The order identifier for the order in this item.
sequence_number_u	uint32_t
	Ignore. Currently not used.
ob_position_u	uint32_t
	Ignore, not used for inactive orders.
combo_mark_c	uint8_t
	Ignore, not used in this query.
filler_3_s	char[3]
	Ignore, used for byte alignment.
party	party_t
	See party_t sub structure below.
order	order_t
	Detailed information on the order.
	See order_t sub structure below.
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.



Variable	Description
display_quantity_i	<ul> <li>int64_t</li> <li>This value should only be considered if the order is an iceberg order. It indicates the original displayed quantity of the order and is therefore the value which the iceberg order will use to re-populate its display_quantity_i field once all its current shown quantity has been traded.</li> <li>ASX Trade will automatically populate this field when the total_volume_i field != 0.</li> </ul>
orig_shown_quantity_i	int64_t The shown quantity of the order when it was originally entered.
orig_total_volume_i	int64_t The total_volume_i value of the order when it was originally entered.
timestamp_in	time_spec_t The time when the order was entered <b>or</b> when it was last modified. See time_spec_t sub structure.
timestamp_created	time_spec_t The time when the order was originally entered. See time_spec_t sub structure.

# 42.5.3 party_t

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

# 42.5.4 order_t

Variable	Description
series	series_t This is the series of the order.
trading_code	trading_code_t Indicates the user who created the order. See trading_code_t sub structure below.
order_var	order_var_t See order_var_t sub structure below.
ex_user_code	ex_user_code_t Indicates the user who last acted on the order. See ex_user_code_t sub structure below.
give_up_member	give_up_member_t



Variable	Description
	Indicates the clearing identity for the order. See give_up_member_t sub structure below.
exchange_info_s	char[32] A free text field used at 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. 98 indicates MO98).
change_reason_c	<ul> <li>uint8_t</li> <li>Indicates the reason for the change. Possible values include:</li> <li>1 = Order deleted</li> <li>3 = Trade</li> <li>4 = Order inactivated</li> <li>5 = Order amended</li> <li>6 = Order added</li> <li>8 = Order price changed</li> <li>9 = Order deleted by trade system</li> <li>10 = Order deleted by proxy</li> <li>13 = Hidden volume order recalculated</li> <li>19 = Trade system deleted day order</li> <li>21 = Inactivated due to Purge</li> <li>22 = Deleted by system due to Instrument Session change</li> <li>22 = Deleted by system due to no shown quantity</li> <li>23 = Inactivated due to Purge</li> <li>24 = Inactivated due to Price away from the market</li> <li>28 = Order reload at normal system start</li> <li>31 = Order reload at intraday Market Place restart</li> <li>34 = Cancelled After Auction</li> <li>39 = Convert undisclosed order to normal order - for active orders falling below minimum order value due to Market Maker protection delta limit reached or exceeded</li> <li>42 = Quote deleted due to Market Maker protection delta limit reached or exceeded</li> <li>43 = Market-to-Limit Sweep order converted to Limit order</li> <li>49 = Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero</li> <li>50 = Sweep order reloaded without MAQ and mid-tick attribute.</li> </ul>
filler_1_s	char[1] Ignore. Used for byte alignment.

# 42.5.5 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.



Variable	Description
premium_i	<pre>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 (order_type_c &gt; 1) Any value = Fixed price order (order_type_c=1)</pre>
	Note that the price for a combination order can be positive, zero or negative.
block_n	uint32_t Block size. Possible values include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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 equity it will be valid for the maximum expiry time for an order.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 (user order_type_c to determine Limit or Market or Limit).</pre>
ex_client_s	char[10]



Variable	Description
	Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report codes. Possible values include: 0 = Ignore, not relevant For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> Information.
order_type_c	<pre>uint8_t Order Type - a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer) 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.</pre>
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 42.5.6 trading_code_t/ex_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 three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.



## 42.5.7 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 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.

# 42.5.8 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.



# 43 MQ78 Query Trade Reports

## 43.1 Query Function

This query is used to query for unmatched trade reports for the participant.

An unmatched trade report is a single-sided trade report (refer to *MO75 Trade Report* in *ASX Trade Transactions*) that has been entered into the system, but not yet matched by a corresponding trade report by the opposing participant

### 43.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_tot_order_t	
Partitioned	True	
Segmented	false	
Answers	MA78	

#### 43.3 Answer Properties

Transaction Type	MA78
Struct Name	answer_trd_report_t

#### 43.4 Message Structure

#### 43.4.1 query_tot_order_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 78}.
series	series_t To retrieve all unmatched trade reports for a participant, zero-fill this series field in the first query. After a successful MQ78 transaction, a number of answer items are returned to the sender. To get the next partition, the Series and Order Index in the previous answer should be used. If the Series in the answer is zero-filled the end of the last partition is reached.
whose	whose_t Used to filter the results. See whose_t sub structure below.
order_index_u	uint32_t Use this field to retrieve information that requires more than one response. On the first query set this field to zero, then on subsequent queries, used the value in the order_index_u field in the answer. A zero value in the order_index_u field indicates the last response.



## 43.4.2 whose_t

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

Type of Trade Report	Fields to be Completed
My Trade Reports	country_id_s, ex_customer_s, user_id_s
Our Trade Reports	country_id_s, ex_customer_s
My Trade Reports for a Specific Client	country_id_s, ex_customer_s, user_id_s ex_client_s
Our Trade Reports for a Specific Client	country_id_s, ex_customer_s and ex_client_s



### Note:

Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] When placing a trade report this field is a free text and is typically used to indicate to participants the ultimate client for this side of the trade. In this structure it can be used as a filter so that the answer contains only those trade reports that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 43.4.3 trading_code_t

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



# 43.5 Answer Structure

# 43.5.1 answer_trd_report_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 78}.
series	series_t The value in this field, along with the order_index_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_index_u	uint32_t The value in this field should be used in the next query. Zero indicates that there is no more information to retrieve.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_trd_report_item_t[290] An array containing the trade report details. See answer_trd_report_item_t sub structure below.

## 43.5.2 answer_trd_report_item_t

Variable	Description
trading_code	trading_code_t This is the actual user that entered this trade report. See trading_code_t sub structure.
transaction_type	transaction_type_t The transaction number that was used to enter this trade report.
order_number_u	quad_word The order identifier for this side of the trade report.
series	series_t This is the series of this side of the trade report.
order_var	order_var_t See order_var_t sub structure.
party	party_t The trading participant who is the opposing party to the trade. This only indicates the participant, not the actual user. See party_t sub structure.
sequence_number_u	uint32_t Ignore. Currently not used.
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



Variable	Description
	Indicates the clearing identifier to be used for this trade report.
	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 As Of date, date part.
	Format: YYYYMMDD.
time_of_agreement_time_s	char[6]
	The As Of time, 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 0.
	For trade reports that include short sell quantity (exch_order_type_n = 2), must be greater
	than 0.

# 43.5.3 trading_code_t

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

# 43.5.4 order_var_t

Variable	Description
mp_quantity_i	int64_t The quantity of the reported trade.
premium_i	int32_t



Variable	Description	
	The price of the reported trade, given to the correct number of decimal points as per the configuration of the instrument class. Users can provide up to four decimal points on the price by restating the whole price in the extended_price_q field in the asx_exchange_info_t struct.	
block_n	uint32_t Block size. Possible values include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.	
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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 equity it will be valid for the maximum expiry time for an order.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>	
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.	
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.	
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.	
open_close_req_c	uint8_t 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 codes. Possible values: 0 = Ignore, not relevant. For other values refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> Information.	



Variable	Description
order_type_c	uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer).
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 43.5.5 party_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "ALI" 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.

## 43.5.6 give_up_member_t

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

# 43.5.7 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 include: 0 = No value



Variable	Description
	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.

## 43.5.8 ASX Specific Overlay of regulatory_data_s Variable

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

Variable	Description	Character Position	ASIC defined content
capacity_of_participant_s char[1]		0	Capacity of participant where: A = Agency P = Principal M = Mixed Agency and Principal.
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False (default).
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.



## 44 MQ80 Unmatched Trade Reports Counterparty Query

## 44.1 Query Function

This query is used where a participant is the counterparty and has not yet reported their side of the trade, but wants to retrieve unmatched trade reports from other participants.

### 44.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_tot_party_t	
Partitioned	True	
Segmented	True	
Answers	MA80	

#### 44.3 Answer Properties

Transaction Type	MA80
Struct Name	answer_trd_report_party_t

## 44.4 Message Structure

## 44.4.1 query_tot_party_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 80}.
series	series_t To retrieve all unmatched trade reports where the own participant is the counterparty, on the first query, zero-fill this series field, the order_number_u field and bid_or_ask_c. On subsequent queries fill in the same three fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field can also act as a filter for a particular instrument series. Fill in the entire structure to indicate that the response should only contain the own unmatched trade reports for that series. The final response is indicated by the order_number_u field in the answer being zero.
order_number_u	quad_word Use this field to retrieve information that requires more than one response. On the first query set this field to zero, then on subsequent queries, used the value in the order_number_u field in the answer. A zero value in the order_number_u field indicates the last response. Can also be used as a filter to query for a specific unmatched trade report.
bid_or_ask_c	uint8_t



Variable	Description
	Use this field to retrieve information that requires more than one response. On the first query set this field to zero, then on subsequent queries, used the value in the bid_or_ask_c field in the answer. Can also be used as a filter to query for a specific unmatched trade report. Possible values: 1 = Buy 2 = Sell.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 44.5 Answer Structure

# 44.5.1 answer_trd_report_party_t

Variable	Description	
transaction_type	transaction_type_t Contains the following: {'M', 'A', 80}.	
series	series_t The value in this field, along with the order_number_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.	
order_number_u	quad_word The value in this field, along with the series value, should be used in the next query. A zero-filled structure indicates this is the last answer.	
items_n	uint16_t The number of items held in the array.	
bid_or_ask_c	uint8_t The value in this field, along with the series value, should be used in the next query.	
filler_1_s	char[1] Ignore. Used for byte alignment.	
item	answer_trd_report_party_item_t[300] An array containing the trade report details. See answer_trd_report_party_item_t sub structure below.	

# 44.5.2 answer_trd_report_item_t

Variable	Description
trading_code	trading_code_t This is the actual user that entered this trade report. See trading_code_t sub structure.
transaction_type	transaction_type_t The transaction_type that was used to add this trade report.
order_number_u	quad_word The order identifier for this side of the trade report.
series	series_t This is the series of this side of the trade report.
order_var	order_var_t



Variable	Description
	See order_var_t sub structure below.
party	party_t The trading participant who is the opposing party to the trade. This only indicates the participant, not the actual user. See party_t sub structure.
sequence_number_u	uint32_t Ignore. Currently not used.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.
give_up_member	give_up_member_t Indicates the clearing identifier to be used for this trade report. 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 As Of date, date part. Format: YYYYMMDD.
time_of_agreement_time_s	char[6] The As Of date, 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.

# 44.5.3 trading_code_t

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



# 44.5.4 order_var_t

Variable	Description
mp_quantity_i	int64_t Quantity of the trade report.
premium_i	int32_t The price of the reported trade, given to the correct number of decimal points as per the configuration of the instrument class. Users can provide up to four decimal points on the price by restating the whole price in the extended_price_q field in the asx_exchange_info_t struct.
block_n	uint32_t Block size. Possible values: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	uint16_t Not used in this query.
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible values include:



Variable	Description
	0 = Ignore, not relevant. For other values see Trade Benort Type in ASX Trade Introduction and Business
	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 include: 1 = Limit price order (premium_i = an integer).
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 44.5.5 party_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a 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.

## 44.5.6 asx_exchange_info_t

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



## 44.5.7 give_up_member_t

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



# 45 MQ92 Own Orders Query

## 45.1 Query Function

This query is used for querying a participant's own orders in the order book. The query can be used to query for both active and central inactive orders.

## 45.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_order_private_filter_t	
Partitioned	True	
Segmented	True	
Answers	MA92	

#### 45.3 Answer Properties

Transaction Type	MA92
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs are described below.

### 45.4 Message Structure

## 45.4.1 query_order_private_filter_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 92}.
series	series_t To retrieve all own orders, on the first query, zero-fill this series field, the order_number_u field and bid_or_ask_c. On subsequent queries users are to fill in the series and order_number_u fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled.
search_series	series_t Acts as a filter for this query. Fill in country and market code to retrieve information on that particular market. Fill in country, market and instrument group code to retrieve information on that particular instrument type. Fill in country, market, instrument group and commodity code to retrieve information on that particular instrument class. Or provide a full series definition to query for a particular instrument series.
whose	whose_t Used to filter the results. See whose_t sub structure below.
order_number_u	quad_word Use this field to retrieve information that requires more than one response. On the first query set this field to zero, then on subsequent queries, used the value in the



Variable	Description
	order_number_u field in the answer. A zero value in the order_number_u field indicates the last response.
order_filter_i	<ul> <li>int32_t</li> <li>Used to filter the results on the types of orders to return.</li> <li>Acts as a bitmap that defines which orders will be returned by the query. Several bits can be combined to return all orders matching the selected bit filters. To query for all types of orders, either set the field empty or set all bits. Possible values:</li> <li>1 = All active orders (excluding Centre Point orders).</li> <li>2 = All active quotes entered with quoting transactions MO36 and MO37</li> <li>4 = All active Centre Point, Centre Point Block and dual-posted Sweep orders</li> <li>8 = All stop orders (not used by ASX)</li> <li>16 = All central inactive orders and quotes (including central inactive Centre Point and Centre Point Block orders).</li> </ul>
bid_or_ask_c	uint8_t Always set to zero in this query.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 45.4.2 whose_t

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

Type of Trade Report	Fields to be Completed
My Orders	country_id_s, ex_customer_s, user_id_s
Our Orders	country_id_s, ex_customer_s
My Orders for a Specific Client	country_id_s, ex_customer_s, user_id_s ex_client_s
Our Orders for a Specific Client	country_id_s, ex_customer_s and ex_client_s



# Note:

Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] This field is a free text and is typically used to indicate to ASX participants the ultimate client for this side of the trade. In this structure it can be used as a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2]



Variable	Description
	Ignore. Used for byte alignment.

### 45.5 Answer Structure

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

- answer_order_hdr_t
- one or more sequences of:
  - item_hdr_t
  - one or more sequences of:
    - sub_item_hdr_t
    - a choice of:
      - basic_order_book_order_t (named structure 34817)
      - reserve_order_t (named structure 34812)
      - 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)
      - ranking_time_t (named structure 34949)
      - crossing_t (named structure 34820)
      - regulatory_t (named structure 34821)
      - short_sell_order_t (named structure 34829).

### 45.5.1 answer_order_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 92}.
next_series	series_t The value in this field should be used in the next query. A zero-filled structure indicates this is the last answer.
next_order_number_u	quad_word The value in this field, along with the series, should be used in the next query.
bid_or_ask_c	uint8_t Ignore. Not used for this query. The next query should always be sent with bid_or_ask_c set to zero.
filler_3_s	char[3] Ignore. Used for byte alignment.
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.



## 45.5.2 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.	

## 45.5.3 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.

## 45.5.4 basic_order_book_order_t (named struct number 34817)

Variable	Description
series	series_t
	The series for which the order exists.
owner	trading_code_t
	Indicates the user who originally placed the order.
	See trading_code_t sub structure below.
give_up_member	give_up_member_t
	The clearing identifier used for the order.
	See give_up_member_t sub structure below.
order number u	quad_word
	The order identifier for the order in this item.
timestamp_in	time_spect_t
. –	The time when the order was entered OR when it was last modified.
	See time_spec_t sub structure.
timestamp_created	time_spec_t
	The time when the order was first entered.
	See time_spec_t sub structure.
sequence_number_u	uint32_t
	Ignore. Currently not used.
ob position u	uint32_t
	Indicates the order book position, 1 being the highest priority.
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 tailor made combination order can be positive, zero or negative.
quantity_i	int64_t
· —	Quantity of the order.



Variable	Description
	For iceberg orders this is 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.
original_quantity_i	int64_t The quantity that the order was originally entered with. For iceberg orders this is the original total quantity of the order. The original 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	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 premium_i = an integer, order_type_c = 65 = Imbalance Limit order) Can also be combined with other exchange specific order types/attributes outlined below. 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 (user order_type_c to determine Market or Limit).</pre>
transaction_number_n	uint16_t



Variable	Description
	The transaction number that was used to originally enter this order (e.g. 1 indicates MO1).
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.
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.
order_type_c	<ul> <li>uint8_t</li> <li>Order type – a combination of this field and the premium_i field signifies different types of orders. Possible values:</li> <li>1 = Limit price order (premium_i = an integer)</li> <li>2 = Market order (premium_i = 0)</li> <li>3 = Market-to-Limit order (premium_i = 0)</li> <li>17 = Best-Limit order (premium_i = 0 and time_validity_n !=0</li> <li>65 = Imbalance Limit order (premium_i = an integer).</li> <li>Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Sweep, Centre Point and Centre Point Block orders.</li> </ul>
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
change_reason_c	<pre>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 trade system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Trade system deleted day order 21 = Inactivated by system due to Instrument Session change 22 = Deleted by system due to on shown quantity 23 = Inactivated due to Purge 24 = Inactivated due to Purge 24 = 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</pre>



Variable	Description
	39 = Convert undisclosed order to normal order - for active orders falling below 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.

# 45.5.5 trading_code_t

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

# 45.5.6 give_up_member_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a 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.

# 45.5.7 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.	



## 45.5.8 reserve_order_t (named structure 34812)

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

## 45.5.9 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.			
mid_tick_c	<ul> <li>uint8_t</li> <li>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).</li> <li>Specifies whether a Limit Sweep order is eligible for passive execution in Centre Point at a half-tick above the limit price.</li> <li>0 = not defined</li> <li>1 = mid-tick attribute set on</li> <li>2 = mid-tick attribute set off</li> <li>3 = allowed for permitted prices in addition to mid-point ('dark limit' order)</li> <li>4 = allowed for permitted prices in addition to mid-point ('dark limit' order), with mid-tick attribute set on</li> <li>5 = Any Price Block order</li> <li>6 = Any Price Block order with mid-tick attribute set to on.</li> <li>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.</li> </ul>			
preference_only_c	uint8_t Specifies whether a Centre Point Order or a Centre Point Block order is a Preference and Kill order or not. For Limit Sweep orders, only '0' or '2' are valid values. 0 = not defined or 'no', order is not a Preference and Kill order. 1 = 'yes', order is a Preference and Kill order. Time validity must be set to Fill and Kill or Fill or Kill for this option. 2 = 'no', order is not a Preference and Kill order. This variable cannot be amended.			
single_fill_minimum_qua y_c	antit 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).			



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

## 45.5.10 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 1 = central inactive order 2 = active order.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 45.5.11 order_submitter_t (named structure 34819)

This structure is included when an order has been entered by ASX Trading Operations on behalf of a participant, showing the user who entered the order.

Variable	Description
submitter	trading_code_t
	See trading_code_t sub structure above.

### 45.5.12 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 above.

### 45.5.13 crossing_t (named structure 34820)

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

## 45.5.14 regulatory_t (named structure 34821)

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



## 45.5.14.1 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 (default).
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.

## 45.5.15 short_sell_order_t (named structure 34829)

Variable	Description
short_sell_quantity_i	<pre>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.</pre>

## 45.5.16 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.		



## 46 MQ99 Maximum Block Order Sizes Query

## 46.1 Query Function

This query is used for requesting information about the maximum number of quotes that can be used in block quotation (MO36) transactions.

After a successful MQ99 transaction, an answer is received with the maximum limits.

### 46.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_block_size_t	
Partitioned	False	
Segmented	False	
Answers	MA99	

### 46.3 Answer Properties

Transaction Type	MA99
Struct Name	answer_block_size_t

### 46.4 Message Structure

### 46.4.1 query_block_size_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 99}.
series	series_t Ignore. Set to zero.

## 46.5 Answer Structure

#### 46.5.1 answer_block_size_t

Variable	Description	
transaction_type	transaction_type_t Contains the following: {'M', 'A', 99}.	
max_block_order_size_i	int32_t Ignore. Not used in ASX Trade.	
max_block_price_size_i	int32_t Maximum items in a two sided price quotation block transaction (MO36). Indicates the number of buy and the number of sell quotations that can be entered in the transaction.	


Variable	Description
	E.g. A value of 10 indicates that a MO36 transaction can contain 10 double sided quotes (10 buy AND 10 sell quotations).



#### 47 MQ151 Query Order Broadcast

#### 47.1 Query Function

This query is used to retrieve missing BO5 order book broadcasts (refer to *BO5 Firm Order Book* in *ASX Trade Broadcasts* for more information).

This transaction should be used after coming on-line to recreate the full BO5 order stream. Each BO5 contains a sequence number. These numbers are unique per Matching Engine partition, per day, per participant.

Users should be aware that the BO5 broadcasts they receive depend on their user type configured in ASX Trade. For instance, a user who is only allowed to access equities and warrants will only receive BO5 broadcasts relating to those types of instruments. Therefore, they will be noticing gaps in the broadcasts sequence numbers and should not infer from these gaps that they have missed broadcasts and hence are required to begin their recovery process.

Furthermore, the MQ151 query relies on the cache of the BO5 broadcasts that are stored at the gateway. Depending on other users who access that gateway, it is possible that the answer will include broadcasts that relate to instruments to which they do not have access.

For instance, if User A only has access to equities and warrants, and User B only has access to derivatives, then the BO5 broadcasts they receive would reflect their system configured rights. However, if both users accessed the same gateway then the BO5 broadcasts are cached on that gateway in the one buffer. If both users were to execute a MQ151 query to retrieve all order history for that day, then the gateway would respond with all the order book broadcasts stored in that one buffer, effectively providing both users with broadcasts that do not concern them. The OI application must handle this situation.

#### 47.2 Query Properties

The facility number is alpha numeric, the range of its value is 0-9 and continues with A-Z, where A equals 10 and B equals 11, etc. For example facility EPA is calculated as EPO+10. Use omniapi_get_info_ex to retrieve the numeric value for EPO (external facility).

Function Call	omniapi_query_ex	
Facility	EPA	
Struct Name	 query_order_broadcast_t	
Partitioned	True	
Segmented	True (using sequence numbers)	
Answers	MA151	

#### 47.3 Answer Properties

Transaction Type	MA151
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs are described below.



## 47.4 Message Structure

#### 47.4.1 query_order_broadcast_t

Variable	Description
transaction type	transaction type t
	Set the structure to the following: {'M', 'Q', 151}.
series	series_t
	Set to zeros to retrieve BO5s from all markets.
	Fill in country_c and market_c to retrieve BO5s from that market. Fill in country_c,
	market_c and instrument_group_c to retrieve BO5s relating to that instrument type.
instance_c	uint8_t
	Indicates that the user wants to retrieve BO5s that originated from this particular
	Matching Engine partition.
	A user should start querying from instance one and query for all BO5s using the sequence
	numbers. In the answer there is a field called instance_next_c that is used to indicate to
	what instance the next query should be directed. This field is set when sequence_last_u is
	set to zero and the last BO5 (for this instance) is sent in the answer. If the last instance has
	been queried, instance_next_c in the answer is set to zero indicating that there are no
	more instances to query.
filler_3_s	char[3]
	Ignore. Used for byte alignment.
yyyymmdd_s	char[8]
	Indicates that the user wants to retrieve BO5s that were issued on this date.
	Format YYYYMMDD.
	Can either be today's date or the previous business day.
broadcast_type	broadcast_type_t
	Set the structure to the following: {'B', 'O', 5}.
sequence_first_u	uint32_t
	Indicates the first sequence number of the BO5 the user wishes to retrieve. Sequence
	numbers for BO5s are issued per Matching Engine partition, per day, per participant. A
	value of zero in this field indicates no lower bound on sequence numbers.
sequence_last_u	uint32_t
	Indicates the last sequence number of the BO5 the user wishes to retrieve. Sequence
	numbers for BO5s are issued per Matching Engine partition, per day, per participant.
	A value of zero in this field indicates no upper bound on sequence numbers. This is used to
	indicate to the system that the user wants all the broadcasts from the partition as per the
	instance_c field, and once retrieved; the user will be directed to the next partition to
	retrieve the broadcasts from that.
	A non-zero value will result in the answer containing only the broadcasts from the
	partition requested as per the instance_c field. Once the user has retrieved these the
	answer will <b>not</b> indicate to the user that they should query for another partition.

#### 47.4.2 Return Codes

An MQ151 query may also be aborted by ASX Trade, in which case only the reason for the transaction being aborted is returned to the sender.



Cstatus	Txstat
Successful	0 – All history items returned.
Transaction Aborted	MP_OHS_DATAPURGED - Order history cache is purged.
Transaction Aborted	ME_OHS_DATAINCOMPLETE - Order history cache is incomplete or inconsistent with respect to this query.
Transaction Aborted	ME_OHS_DATAINCOMPLETE_NORECOV - Order history cache is incomplete and the gateway cannot recover it from the central system.

#### 47.5 Answer Structure

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

- answer_hdr_t
- one or more sequences of:
  - item_hdr_t
  - one or more sequences of:
    - sub_item_hdr_t
    - a choice of:
      - query_order_broadcast_next_t (named structure 34911)
      - 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 first item in the answer will always be the query_order_broadcast_next_t structure, indicating to the user if there are any more broadcasts to retrieve and if so, how to retrieve them.

An example of retrieving all missing BO5s for the current date is as follows:

- 1. In the query struct, set series field to zero, instance_c to 1, yyyymmdd_s to today's date, broadcast type to {'B', 'O', 5}, sequence_first_u to 0 and sequence_last_u to 0.
- 2. On subsequent queries, set series field to zero, instance_c to the value received in instance_next_c in the query_order_broadcast_next_t structure, broadcast type to {'B', 'O', 5}, sequence_first_u to the value received in sequence_first_next_u in the query_order_broadcast_next_t structure and sequence_last_u to 0.
- 3. Repeat step 2 until the value for instance_next_c in the query_order_broadcast_next_t structure is zero.

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 151}.
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.

#### 47.5.1 answer_hdr_t

#### 47.5.2 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.

#### 47.5.3 sub_item_hdr_t

Variable	Description
named_struct_n	uint16_t



Variable	Description
	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.

## 47.5.4 query_order_broadcast_next_t (named structure 34911)

Variable	Description
sequence_first_next_u	uint32_t Use the value for the next query in the field sequence_first_u.
instance_next_c	uint8_t Use the value in this field for the next query in the field instance_c. When this is zero then all answer data has been received.
filler_3_s	char[3] Ignore. Used for byte alignment.

Contains the information necessary to retrieve the next segment of the MA151.

## 47.5.5 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 below.
exchange_info_s	char[32] A free text field used at the participant's discretion.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to 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 below.
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 below.

## 47.5.6 block_price_trans_item_t

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



Variable	Description
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. <b>Note</b> : it is possible for a price to be zero, 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. <b>Note</b> : 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 include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>The time validity that applies to both orders in this item - on the bid and the ask side.</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>



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. <b>Note</b> : 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 bytes 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) Best of day order inactivated due to purge
order_type_c	<ul> <li>uint8_t</li> <li>Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include:</li> <li>1 = Limit price order (premium_i = an integer)</li> <li>2 = Market order (premium_i = 0)</li> <li>3 = Market-to-Limit order (premium_i = 0)</li> <li>17 = Best-Limit order (premium_i = 0, time_validity_n != 0)</li> <li>65 = Imbalance Limit order (premium_i = an integer).</li> <li>Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Centre Point, Centre Point Block and Sweep orders.</li> </ul>
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
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 values: 1 = Absolute quantity.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 47.5.7 hv_alter_trans_t (named structure 34010)

When an order is amended using a transaction that involves the hv_alter_trans_t, 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 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



Variable	Description
	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.
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 participant's discretion.
total_volume_i	<pre>int64_t The total quantity of the order, i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
delta_quantity_c	uint8_t Usually, on an order amendment, this field would indicate 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 include: 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 values include: 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.

# 47.5.8 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 preamble for that struct.

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.



Variable	Description
order_var	order_var_t See order_var_t sub structure.
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 participant's discretion.
total_volume_i	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
delta_quantity_c	uint8_t Usually, on an order amendment, this field would indicate 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 include: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.
balance_quantity_i	<ul> <li>int64_t</li> <li>Indicates that the user wants a check done on the quantity of the existing order before amendment. Possible values include:</li> <li>0 = No balance check is performed</li> <li>Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.</li> </ul>

# 47.5.9 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.
series	series_t The series to which this order information relates.
order_var	order_var_t See order_var_t sub structure.
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 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 0.



Variable	Description	
	Normal order: total_volume_i = 0, display_quantity_i != 0	
	Iceberg order: total_volume_i = display_quantity_i != 0	
	<pre>Iceberg order: total_volume_i &gt; display_quantity_i != 0.</pre>	

## 47.5.10 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 trading_code_t sub structure.
order_var	order_var_t See order_var_t sub structure.
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 participant's discretion.
total_volume_i	<pre>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 &gt; 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 &gt; display_quantity_i != 0.</pre>

# 47.5.11 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 The series to which this order information relates.
give_up_member	give_up_member_t See give_up_member_t 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. <b>Note</b> : It is possible for a price to be zero, indicating a market order, or even negative for a combination.
ask_premium_i	int32_t



Variable	Description
	Indicates the price of the order on the ask side of this block transaction. <b>Note</b> : 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 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 include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>The time validity that applies to both orders in this block transaction - on the bid and the ask side.</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
order_type_c	uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 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 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 participants their own order identifier.
exchange_info_s	char[32] A free text field used at participant's discretion.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable.

#### 47.5.12 multi_order_response_t (named structure 34906)

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

Variable	Description
transaction_status_i	<pre>int32_t Contains the same information as returned from the omniapi_tx_ex function, indicating the status of transaction. Possible values include: 0 = Successful Any other value = Transaction aborted.</pre>
trans_ack_i	<ul> <li>int32_t</li> <li>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:</li> <li>1 = No part of the order placed in the order book and no part closed</li> <li>2 = The whole order closed</li> <li>3 = The order partially closed and nothing placed in the order book</li> <li>4 = The whole order placed in the order book</li> <li>6 = The order partially placed in the order book and partially closed.</li> <li>GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type.</li> <li>MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.</li> </ul>
item_number_c	uint8_t The item number in a quote transaction with a value of zero representing the first item. Note that 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.

#### 47.5.13 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 struct is sent in the same broadcast.





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	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
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 include: 1 = Bid



Variable	Description
	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 trade system
	10 = Order deleted by proxy
	13 = Hidden volume order recalculated
	19 = Trade system deleted day order
	21 = Inactivated by system due to Instrument Session change
	22 = Deleted by system due to no shown quantity
	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.

## 47.5.14 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.
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.



Variable	Description
total_volume_i	<pre>int64_t The total quantity of the order, i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 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 anended 6 = Order amended 8 = Order price changed 9 = Order deleted by trade system 10 = Order deleted by trade system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Trade system deleted day order 21 = Inactivated by system due to Instrument Session change 22 = Deleted by system due to no shown quantity 23 = Inactivated due to Purge 24 = Inactivated due to Purge 24 = Inactivated due to Expiry 27 = Inactivated due to Expiry 27 = Inactivated due to Price away from the market 28 = Order reload at normal system start 31 = 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 43 = 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 participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.



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.

## 47.5.15 order_chg_sep_trans_ack_t (named structure 34919)

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

# 47.5.16 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.
order	order_t See order_t sub structure.
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.

## 47.5.17 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 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. 31 indicates MO31).
change_reason_c	<ul> <li>uint8_t</li> <li>Indicates the reason for the change. Possible values include:</li> <li>1 = Order deleted</li> <li>3 = Trade</li> <li>4 = Order inactivated</li> <li>5 = Order amended</li> <li>6 = Order added</li> <li>8 = Order price changed</li> <li>9 = Order deleted by trade system</li> <li>10 = Order deleted by proxy</li> <li>13 = Hidden volume order recalculated</li> <li>19 = Trade system deleted day order</li> <li>21 = Inactivated by system due to Instrument Session change</li> <li>22 = Deleted by system due to no shown quantity</li> <li>23 = Inactivated due to Purge</li> <li>24 = Inactivate day orders</li> </ul>



Variable	Description
	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.

# 47.5.18 order_price_change_t (named structure 34905)

The order_price_change_t 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
	Indicates the reason for the change. Possible values include:
	1 = Order deleted
	3 = Trade
	4 = Order inactivated
	5 = Order amended
	6 = Order added
	7 = Market Order converted to a limit order
	8 = Order price changed
	9 = Order deleted by trade system
	10 = Order deleted by proxy
	13 = Hidden volume order recalculated
	19 = Trade system deleted day order



Variable	Description
	21 = Inactivated by system due to Instrument Session change
	22 = Deleted by system due to no shown quantity
	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_2_s	char[2]
	Ignore. Used for byte alignment.

## 47.5.19 order_return_info_t (named structure 34904)

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

Variable	Description
trans_ack_i	<ul> <li>int32_t</li> <li>The answer to the user. Contains the transaction status returned in the omniapi_tx_ex function.</li> <li>Return codes vary depending on the context in which they occur, but some common examples would be:</li> <li>1 = No part of the order placed in the order book and no part closed</li> <li>2 = The whole order closed</li> <li>3 = The order partially closed and nothing placed in the order book</li> <li>4 = The whole order placed in the order book</li> <li>6 = The order partially placed in the order book and partially closed</li> <li>GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type.</li> <li>MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low Order Book volume.</li> </ul>
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.



## 47.5.20 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 14Kb.

Variable	Description
segment_number_n	uint16_t A BO5 broadcast can be segmented if the one broadcast contains too much information to be sent at once. This field will indicate the number of the segment of the BO5, 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 BO5 broadcasts, the value in this field is zero.
instance_c	uint8_t Indicates from which Matching Engine 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 1 for each segment.
trading_code	trading_code_t See trading_code_t sub struct below.

## 47.5.21 order_trade_info_t (named structure 34920)

Variable	Description
match_id	match_id_t
	Ignore. Not used.
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 Trade Source in <i>ASX Trade Introduction and Business</i> Information 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.



## 47.5.22 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 Ignore. Not used.
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 Trade Source in ASX Trade Introduction and Business Information 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.

# 47.5.23 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 <b>AND</b> operator can be used on the above possible values. Refer to <u>ASX Trade Markets, Instrument Groups and Trade Condition Codes</u> .
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 4096 = Centre Point Block order.



Variable	Description
ext_t_state_c	uint8_t
	Trade report code. Possible values:
	0 = Ignore, not relevant.
	For other values see Trade Report Types in ASX Trade Introduction and Business
	Information 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 maker) = 0 (zero)
	Aggressive (price taker) = 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 noncrossed
	trades, zero will be returned.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

## 47.5.24 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.
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_exchange_info_t sub structure.
give_up_member	give_up_member_t See give_up_member_t sub structure.
settlement_date_s	char[8]



Variable	Description
	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.
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	<pre>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.</pre>

#### 47.5.25 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.
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_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	<pre>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.</pre>

## 47.5.26 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 codes. Possible values include: 0 = Ignore, not relevant. For other values refer to <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> <i>Information</i> .



Variable	Description
deferred_publication_c	uint8_t Ignore. Not used by ASX Trade.
bid_side	trd_rpt_cust_t See trd_rpt_cust_t sub structure.
ask_side	trd_rpt_cust_t See trd_rpt_cust_t sub structure.

## 47.5.27 trd_rpt_cust_t

Variable	Description
party	party_t This is the declared counterparty for this part of the trade. See party_t sub structure below.
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure below.
open_close_req_c	uint8_t Ignore. Currently not used.
exch_order_type_n	uint16_t Exchange specific order types. Possible values: 2 = Short Sell order.
give_up_member	give_up_member_t See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable.
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.

## 47.5.28 exchange_info_t (named structure 50004)

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



## 47.5.29 free_text_t (named structure 34801)

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

## 47.5.30 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.

## 47.5.31 single_order_insert (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.

## 47.5.32 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



Variable	Description
	2 = Sell.
filler_3_s	char[3] Ignore. Used for byte alignment.
items_n	uint16_t The number of sub-items following this top structure.
size_n	uint16_t The total size of the message, including this header.

#### 47.5.33 basic_order_t (named structure 34810)

Variable	Description
premium_i	<pre>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 tailor made combination order can be positive, zero or negative.</pre>
quantity_i	int64_t Quantity of the order. For iceberg orders this is 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	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
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



Variable	Description
	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)
	Can also be combined with other exchange specific order types/attributes) outlined below. 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 $n \neq 0$ ).
	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 (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).
order_type_c	uint8_t Order type – a combination of this field and the premium_i field signifies different types of orders. Possible values:
	1 = Limit price order (premium_i = an integer)
	2 = Market order (premium_i = 0)
	3 = Market-to-Limit order (premium_i = 0)
	17 = Best-Limit order (premium_i = 0 and time_validity_n !=0
	65 = Imbalance Limit order (premium_i = an integer).
	Used in conjunction with values in the field exch_order_type_n to determine Undisclosed,
	Sweep, Centre Point and Centre Point Block orders.
bid_or_ask_c	uint8_t
	Bid or Ask. Possible values:
	1 = Buy
	2 = Sell.
filler_2_s	char[2]
	Ignore. Used for byte alignment.

# 47.5.34 reserve_order_t (named structure 34812)

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

## 47.5.35 basic_order_update_t (named structure 34815)

Variable	Description
premium_i	int32_t The price of the order. A combination of this field and the order_type_c field signify different types of orders.



Variable	Description
	0 = Market order.
	Any other value than zero = Limit order (order_type_c = 1 or 65).
	The price for a tailor made combination order can be positive, zero or negative.
quantity_i	int64_t
	Quantity of the order.
	When amending an order, this field can be set to:
	the new value (delta_quantity_c = 1)
	the amount by which it should be changed (delta_quantity_c = 2)
	zero to indicate 'no change'.
	For iceberg orders this is the total quantity of the order. The shown quantity portion of the
	iceberg order is defined in the reserve_order_t sub-structure. For iceberg orders, an
	increase to total quantity is not allowed.
balance_quantity_i	int64_t
	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.
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.
	the day.
	Unit = 2, Count = 0 (i.e. binary 10 0000 0000, nex 200, dec 512): Order is valid until the
	it will be valid for the maximum allowed time for that particular instrument type
	I = 5 Count = a positive integer (i.e. bipary 101 0000 0011 bey 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
	(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)
	Can also be combined with other exchange specific order types / attributes) outlined
	below.
	4 = Market Bid order



Variable	Description
	<pre>(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 (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)</pre>
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 values: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 47.5.36 centre_point_order_t (named structure 34816)

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



Variable	Description
	Specifies whether a Centre Point Order or a Centre Point Block order is a Preference and
	Kill order or not. For Limit Sweep orders, only '0' or '2' are valid values.
	0 = not defined or 'no', order is not a Preference and Kill order.
	1 = 'yes', order is a Preference and Kill order. Time validity must be set to Fill and Kill or Fill
	or Kill for this option.
	2 = 'no', order is not a Preference and Kill order.
	This variable cannot be amended.
single_fill_minimum_qu	uantit uint8_t
y_c	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.

#### 47.5.37 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
	1 = central inactive order
	2 = active order.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

# 47.5.38 order_submitter_t (named structure 34819)

Variable	Description
submitter	trading_code_t See trading_code_t sub structure.

## 47.5.39 order_owner_t (named structure 34804)

Variable	Description
owner	trading_code_t See trading_code_t sub structure.

## 47.5.40 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.



#### 47.5.41 crossing_t (named structure 34820)

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

#### 47.5.42 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.

# 47.5.43 short_sell_order_t (named structure 34829)

Variable	Description
short_sell_quantity_i	<pre>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.</pre>

#### 47.5.44 short_sell_order_change_t (named structure 34830)

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

#### 47.5.45 enhanced_cp_matching_t (named structure34831)

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

#### 47.5.46 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]



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

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

# 47.5.48 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.
premium_i	<pre>int32_t Premium - the price of the order. A combination of this field and the order_type_c field signify different types of orders 0 = Market order (order_type_c &gt; 1) Any value = Fixed price order (order_type_c=1). Note that the premium for a tailor made combination order can be positive, zero or negative.</pre>
block_n	uint32_t Block size. Possible values include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	uint16_t This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte). Unit = 0, Count = 0 (i.e. binary = 0): Order is an "Immediate" type order. Fill Or Kill when block_n = 0. Fill And Kill when block_n = 1. Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day. Unit = 2, Count = 0 (i.e. binary 10 0000 0000, hex 200, dec 512): Order is valid until the instrument expires. Since equities do not expire then if this order relates to an equity it will be valid for the maximum expiry time for an order.



Variable	Description
	<ul> <li>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.</li> <li>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. 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.</li> </ul>
exch_order_type_n	<pre>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</pre>
	<ul> <li>(premium_i = an integer, order_type_c = 1, only entered by ASX Trading Operations).</li> <li>8 = Price Stabilisation/Green Shoe Order</li> <li>(premium_i = an integer, order_type_c = 1, time_validity_n != 0).</li> <li>32 = Undisclosed (use order_type_c to determine order type)</li> <li>64 = Centre Point Order (use order_type_c to determine Market or Limit)</li> <li>2048 = Sweep order (user order_type_c to determine Limit or Market-to-limit)</li> <li>4096 = Centre Point Block order (use order_type_c to determine Market or Limit).</li> </ul>
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report codes. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> 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 include: 1 = Limit price order (premium_i = an integer)



Variable	Description
	2 = Market order (premium_i = 0)
	3 = Market-to-Limit order (premium_i = 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.
stop_condition_c	uint8_t
	Ignore. Currently not used.
filler_2_s	char[2]
	Ignore. Used for byte alignment.

# 47.5.49 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.

# 47.5.50 trading_code_t

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

# 47.5.51 originator_trading_code_t

Variable	Description
country_id_s	char[2] For ASX Trade this is always set to "AU", indicating the Australian exchange.
ex_customer_s	char[5] This is the unique identifier assigned to a customer of the exchange. For trading participants this is typically a three digit number whereas information venders have an


Variable	Description
	alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of an ASX Trade user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

#### 47.5.52 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.

#### 47.5.53 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 include: 0 = No value 1 = Initial trade report 2 = No initial trade report.		
filler_1_s	char[1] Ignore. Used for byte alignment.		
extended_price_q	int64_i This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the special value indicating that it is to be ignored. If the 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.		

### 47.5.54 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.



Variable	Description	Character Position	ASIC defined content
directed_wholesale_s	char[1]	1	Directed wholesale indicator for agency orders and transactions where: Y = True N =False (default).
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.



#### 48 MQ154 Query Order Broadcast on Behalf

#### 48.1 Query Function

This query is used to retrieve missing BO5 order book broadcasts for a specified participant (refer to *BO5 Firm Order Book* in *ASX Trade Broadcasts* for more information). Broker Service Providers need to use this query to be able to retrieve order history for their customers.

This transaction should be used after coming on-line to recreate the full BO5 order stream. Each BO5 contains a sequence number. These numbers are unique per Matching Engine partition, per day, per participant.

Users who request data for a subset of instrument types or classes, or are not permitted to see all instrument types or classes will notice there are gaps in their sequence numbers.

As its counterpart query - the MQ151 - this query accesses the cache of BO5s held at the gateway.

#### 48.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPB	
Struct Name	query_order_broadcast_p_t	
Partitioned	True	
Segmented	True (using sequence numbers)	
Answers	MA154	

#### 48.3 Answer Properties

Function Call	MA154
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the sequence of possible structs are described below.

#### 48.4 Message Structure

#### 48.4.1 query_order_broadcast_p_t

Variable	Description		
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 154}.		
party	party_t The identity of the participant on whose behalf the request is made. See party_t sub structure below.		
series	series_t Set to zeros to retrieve BO5s from all markets. Fill in country_c and market_c to retrieve BO5s from that market. Fill in country_c, market_c and instrument_group_c to retrieve BO5s relating to that instrument type.		
instance_c	uint8_t		



Variable	Description		
	Indicates that the user wants to retrieve BO5s that originated from this particular Matching Engine partition. A user should start querying from instance one and query for all BO5s using the sequence numbers. In the answer there is a field called instance_next_c that is used to indicate to what instance the next query should be directed. This field is set when sequence_last_u is set to zero and the last BO5 (for this instance) is sent in the answer. If the last instance has been queried, instance_next_c in the answer is set to zero indicating that there are no more instances to query.		
filler_3_s	char[3] Ignore. Used for byte alignment.		
yyyymmdd_s	char[8] Indicates that the user wants to retrieve BO5s that were issued out on this date. Format YYYYMMDD. Can either be today's date or the previous business day.		
broadcast_type	broadcast_type_t Set the structure to the following: {'B', 'O', 5}.		
sequence_first_u	uint32_t Indicates the first sequence number of the BO5 the user wishes to retrieve. Sequence numbers for BO5s are issued per Matching Engine partition, per day, per participant. A value of zero in this field indicates no lower bound on sequence numbers.		
sequence_last_u	uint32_t Indicates the last sequential number of the BO5 the user wishes to retrieve. Sequence numbers for BO5s are issued per Matching Engine partition, per day, per participant. A value of zero in this field indicates no upper bound on sequence numbers. This is used to indicate to the system that the user wants all the broadcasts from the partition as per the instance_c field, and once retrieved; the user will be directed to the next partition to retrieve the broadcasts from that. A non-zero value will result in the answer containing only the broadcasts from the partition requested as per the instance_c field. Once the user has retrieved these the answer will <b>not</b> indicate to the user that they should query for another partition.		

## 48.4.2 party_t

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



#### 48.4.3 Return Codes

An MQ154 query may also be aborted by the Marketplace, in which case only the reason for the transaction being aborted is returned to the sender.

Cstatus	Txstat
Successful	0 – All history items returned.
Transaction Aborted	MP_OHS_DATAPURGED – Order history cache is purged.
Transaction Aborted	ME_OHS_DATAINCOMPLETE— Order history cache is incomplete or inconsistent with respect to this query.
Transaction Aborted	ME_OHS_DATAINCOMPLETE_NORECOV – Order history cache is incomplete and the gateway cannot recover it from the central system.

#### 48.5 Answer Structure

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

- answer_hdr_t
- one or more sequences of:
  - item_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 structure34831)
- 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 first item in the answer will always be the query_order_broadcast_next_t structure, indicating to the user if there are any more broadcasts to retrieve and if so, how to retrieve them.

An example of retrieving all missing BO5s for the current date is as follows:

- 1. In the query struct, set series field to zero, instance_c to 1, yyyymmdd_s to today's date, broadcast type to {'B', 'O', 5}, sequence_first_u to 0 and sequence_last_u to 0..
- 2. On subsequent queries, set series field to zero, instance_c to the value received in instance_next_c in the query_order_broadcast_next_t structure, broadcast type to {'B', 'O', 5}, sequence_first_u to the value received in sequence first next u in the query order broadcast next t structure and sequence last u to 0.
- 3. Repeat step 2 until the value for instance_next_c in the query_order_broadcast_next_t structure is zero.

#### 48.5.1 answer_hdr_t

Variable	Description	
transaction_type	transaction_type_t Contains the following: {'M', 'A', 154}.	
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.	

#### 48.5.2 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



#### 48.5.3 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.

### 48.5.4 query_order_broadcast_next_t (named structure 34911)

This contains the information necessary to retrieve the next segment of the MA154.

Variable	Description
sequence_first_next_u	uint32_t Use the value for the next query in the field sequence_first_u.
instance_next_c	uint8_t Use the value in this field for the next query in the field instance_c. When this is zero then all answer data has been received.
filler_3_s	char[3] Ignore. Used for byte alignment.

#### 48.5.5 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.
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 below.
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.



#### 48.5.6 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. <b>Note</b> : 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. <b>Note</b> : 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 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 of the order on the bid side of this item in a block transaction.
ask_total_volume_i	int64_t Indicates the total 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) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>The time validity that applies to both orders in this item - on the bid and the ask side.</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>



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. <b>Note</b> : 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.
order_type_c	<pre>uint8_t 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.</pre>
ex_client_s	char[10] Client – a free text field typically used to indicate to the participant the ultimate client making the order.
delta_quantity_c	uint8_t Usually, on an order amendment, this field would indicate 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: 1 = Absolute quantity.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 48.5.7 hv_alter_trans_t (named structure 34010)

When an order is amended using a transaction that involves the hv_alter_trans_t, 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 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



Variable	Description
	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.
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 participant's discretion.
total_volume_i	<pre>int64_t The total quantity of the order i.e. both the hidden and shown display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
delta_quantity_c	uint8_t Usually, on an order amendment, this field would indicate 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 include: 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 values include: 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.

## 48.5.8 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 preamble for that struct.

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.



Variable	Description
order_var	order_var_t See order_var_t sub structure.
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 participant's discretion.
total_volume_i	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
delta_quantity_c	uint8_t Usually, on an order amendment, this field would indicate 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 include: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.
balance_quantity_i	<ul> <li>int64_t</li> <li>Indicates that the user wants a check done on the quantity of the existing order before amendment. Possible values include:</li> <li>0 = No balance check is performed</li> <li>Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.</li> </ul>

# 48.5.9 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.
series	series_t The series to which this order information relates.
order_var	order_var_t See order_var_t sub structure.
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 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 0



Variable	Description	
	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.	

### 48.5.10 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 trading_code_t sub structure.
order_var	order_var_t See order_var_t sub structure.
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 participant's discretion.
total_volume_i	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>

### 48.5.11 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 The series to which this order information relates.
give_up_member	give_up_member_t See give_up_member_t 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



Variable	Description
	Indicates the price of the order on the ask side of this block transaction. <b>Note</b> : 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	<ul> <li>uint16_t</li> <li>The time validity that applies to both orders in this block transaction - on the bid and the ask side.</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
ex_client_s	char[10] Client – a free text field typically used to indicate to participants the ultimate client making the order.
order_type_c	uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 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 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 participant's discretion.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable.

### 48.5.12 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 include: 0 = Successful Any other value = Transaction aborted.
trans_ack_i	<ul> <li>int32_t</li> <li>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:</li> <li>1 = No part of the order placed in the order book and no part closed</li> <li>2 = The whole order closed</li> <li>3 = The order partially closed and nothing placed in the order book</li> <li>4 = The whole order placed in the order book</li> <li>6 = The order partially placed in the order book and partially close.</li> <li>GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type.</li> <li>MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low order book volume.</li> </ul>
item_number_c	uint8_t The item number in a quote transaction with a value of 0 representing the first item. <b>Note</b> That 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.

### 48.5.13 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 struct 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).

# **(i)**

### 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	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
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 include:



Variable	Description
	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 trade system
	10 = Order deleted by proxy
	13 = Hidden volume order recalculated
	19 = Trade system deleted day order
	21 = Inactivated by system due to Instrument Session change
	22 = Deleted by system due to no shown quantity
	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
	42 = Quote deleted due to market march protection quantity minit reached or exceeded 48 = Market-to-Limit Sween order converted to Limit order
	49 = Centre Point Block or Sweep order bas traded below its MAO and MAO is reset to
	50 = Sweep order reloaded without MAQ and mid-tick attribute.
filler 1 s	char[1]
	Ignore. Used for byte alignment.

#### 48.5.14 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 size and total size fields describe the remaining volumes of the order.

Variable	Description
series	series_t The series to which this order information relates.
order_number_u	quad_word Indicates the identifier of the order.
mp_quantity_i	int64_t



Variable	Description
	Indicates the remaining quantity of the order, after this change was done. A zero indicates the whole order was deleted.
total_volume_i	<pre>int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) quantity. Only set as &gt; 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 &gt; display_quantity_i != 0.</pre>
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 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 anended 6 = Order added 8 = Order price changed 9 = Order deleted by trade system 10 = Order deleted by trade system 10 = Order deleted by proxy 13 = Hidden volume order recalculated 19 = Trade system deleted day order 21 = Inactivated by system due to Instrument Session change 22 = Deleted by system due to no shown quantity 23 = Inactivated due to Purge 24 = Inactivated due to Purge 24 = Inactivated due to Purge 25 = Inactivated due to Purge 26 = Inactivated due to Price away from the market 27 = Inactivated due to Price away from the market 28 = Order reload at normal system start 31 = Order reload at normal system start 31 = Order reload at normal system start 31 = Order reload at normal system start 34 = Cancelled After Auction 39 = Convert undisclosed order to normal order - for active orders falling below the minimum order value due to Market Maker protection delta limit reached or exceeded 42 = Quote deleted due to Market Maker protection quantity limit reached or exceeded 42 = Oucte deleted due to Market Maker protection quantity limit reached or exceeded 43 = 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 participants the ultimate client making
customer_info_s	char[15]



Variable	Description
	Customer information – a free text field typically used to indicate to participants their own order identifier.
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.

# 48.5.15 order_chg_sep_trans_ack_t (named structure 34919)

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

# 48.5.16 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.
order	order_t
	See order_t sub structure.
total_volume_i	int64_t
	Current total volume of the order.
display quantity i	int64_t
	Current shown volume of the order.



Variable	Description
orig_total_volume_i	int64_t The total quantity of the order when it was originally entered.
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.

### 48.5.17 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 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
	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
	22=Deleted by system due to no shown quantity
	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	
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	
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	
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	
30=Order reload at normal system start 31=Order reload at intraday Market Place restart 34=Cancelled After Auction	
31=Order reload at intraday Market Place restart 34=Cancelled After Auction	
34=Cancelled After Auction	
39 = Convert undisclosed order to normal order - for active orders to minimum order value due to trading	falling below the
41 = Quote deleted due to Market Maker protection delta limit rea 42 = Quote deleted due to Market Maker protection quantity limit 48 = Market-to-Limit Sweep order converted to Limit order	ched or exceeded reached or exceeded
49 = Centre Point Block or Sweep order has traded below its MAQ a zero	and MAQ is reset to
50 = Sweep order reloaded without MAQ and mid-tick attribute.	
filler_1_s char[1]	

### 48.5.18 order_price_change_t (named structure 34905)

The order_price_change_t 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
	Indicates the reason for the change. Possible values include:
	1 = Order deleted
	3 = Trade
	4 = Order inactivated
	5 = Order amended
	6 = Order added
	7 = Market Order converted to a limit order
	8 = Order price changed
	9 = Order deleted by trade system
	10 = Order deleted by proxy
	13 = Hidden volume order recalculated



Variable	Description
	19 = Trade system deleted day order
	21 = Inactivated by system due to Instrument Session change
	22 = Deleted by system due to no shown quantity
	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 Sween 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.

### 48.5.19 order_return_info_t (named structure 34904)

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

Variable	Description
trans_ack_i	<ul> <li>int32_t</li> <li>The answer to the user. Contains the transaction status returned in the omniapi_tx_ex function.</li> <li>Return codes vary depending on the context in which they occur, but some common examples would be:</li> <li>1 = No part of the order placed in the order book and no part closed</li> <li>2 = The whole order closed</li> <li>3 = The order partially closed and nothing placed in the order book</li> <li>4 = The whole order placed in the order book</li> <li>6 = The order partially placed in the order book and partially closed</li> <li>GEN_CDC_INT_CLOSED = Instrument type not open for this transaction type</li> <li>MP_MATCH_LOW_VOLUME = Fill or Kill order could not be filled because of low Order Book volume.</li> </ul>
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.



#### 48.5.20 enhanced_cp_matching_t (named structure34831)

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

#### 48.5.21 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 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 Matching Engine 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 below.

#### 48.5.22 order_trade_info_t (named structure 34920)

Variable	Description
match_id	match_id_t
	Ignore. Not used.
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 Trade Source in ASX Trade Introduction and Business
	Information for the possible values in this field.



Variable	Description
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.

# 48.5.23 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 Ignore. Not used.
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</i> Information 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.

### 48.5.24 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 <b>AND</b> 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



Variable	Description
	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
	4096 = Centre Point Block order.
ext_t_state_c	uint8_t
	Trade report code. Possible values:
	0 = Ignore, not relevant.
	For other values see Trade Report Types in ASX Trade Introduction and Business
	Information 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 maker) = 0 (zero)
	Aggressive (price taker) = 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 noncrossed trades, zero will be returned.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

# 48.5.25 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.
party	party_t This is the declared counterparty for the other side of the trade. See party_t sub structure.



Variable	Description
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.
give_up_member	give_up_member_t 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.
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	<pre>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.</pre>

# 48.5.26 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.
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]



Mantalala	Description
Variable	Description
	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]
	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 0.
	For trade reports that include short sell quantity (exch_order_type_n = 2), must be greater than 0.

### 48.5.27 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]



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.
ext_t_state_c	uint8_t Trade report code. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> <i>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.
ask_side	trd_rpt_cust_t See trd_rpt_cust_t sub structure.

### 48.5.28 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 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 participants their own order identifier.
exchange_info_s	char[32] This field is overlaid with asx_exchange_info_t sub structure. The struct is 32 bytes in size, filling the entire field. See asx_exchange_info_t sub structure.
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 in the list below: 2 = Short Sell order
give_up_member	give_up_member_t See give_up_member_t sub structure below.
regulatory_data_s	char[44] Contains regulatory data that must be supplied for each order and transaction. See ASX specific overlay of regulatory_data_s variable.
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.	Variable	Description
For trade reports that include short sell quantity (exch_order_type_n = 2), must be ginter than 0.		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.

### 48.5.29 exchange_info_t (named structure 50004)

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

#### 48.5.30 free_text_t (named structure 34801)

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

#### 48.5.31 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.

#### 48.5.32 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.



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

# 48.5.34 basic_order_t (named structure 34810)

Variable	Description
premium_i	<pre>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 tailor made combination order can be positive, zero or negative.</pre>
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.



Variable	Description
	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	<pre>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 premium_i = an integer, order_type_c = 65 = Imbalance Limit order). Can also be combined with other exchange specific order types/attributes) outlined below. 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 (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).</pre>
order_type_c	<pre>uint8_t Uint8_t Order type - a combination of this field and the premium_i field signifies different types of orders. Possible values: 1 = Limit price order (premium_i = an integer) 2 = Market order (premium_i = 0) 3 = Market-to-Limit order (premium_i = 0) 17 = Best-Limit order (premium_i = 0 and time_validity_n !=0 65 = Imbalance Limit order (premium_i = an integer). Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Sweep, Centre Point and Centre Point Block orders.</pre>
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 48.5.35 reserve_order_t (named structure 34812)

Variable	Description
display_quantity_i	int64_t



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

# 48.5.36 basic_order_update_t (named structure 34815)

Variable	Description
premium_i	<pre>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 tailor made combination order can be positive, zero or negative.</pre>
quantity_i	<pre>int64_t Quantity of the order. When amending an order, this field can be set to: The new value (delta_quantity_c = 1) The amount by which it should be changed (delta_quantity_c = 2) Zero to indicate 'no change'. For iceberg orders this the total quantity of the order. The shown quantity portion of the iceberg order is defined in the reserve_order_t sub-structure. For iceberg orders, an increase to total quantity is not allowed.</pre>
balance_quantity_i	<ul> <li>int64_t</li> <li>Indicates that the participant wants a check done on the quantity of the existing order before amendment. Possible values:</li> <li>0 = No balance check is performed</li> <li>Positive integer = Existing quantity on the order must be the same as this value otherwise the transaction will be rejected.</li> </ul>
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>



Variable	Description
	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	<pre>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</pre>
	<pre>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). Can also be combined with other exchange specific order types/attributes) outlined below. 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)</pre>
	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).
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 values: 1 = Absolute quantity.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 48.5.37 centre_point_order_t (named structure 34816)

Variable	Description
minimum_quantity_i	<ul> <li>int64_t</li> <li>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.</li> <li>0 = no minimum acceptable quantity.</li> <li>Must be set to zero for Centre Point orders (exch. order, type, n = 64 or 66).</li> </ul>
mid_tick_c	<ul> <li>uint8_t</li> <li>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).</li> <li>Specifies whether a Limit Sweep order is eligible for passive execution in Centre Point at a half-tick above the limit price.</li> <li>0 = not defined</li> <li>1 = mid-tick attribute set on</li> <li>2 = mid-tick attribute set off</li> </ul>



Variable	Description
	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.
	This variable cannot be amended.
single_fill_minimum_quan	tit uint8_t
y_c	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.

### 48.5.38 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 1 = central inactive order 2 = active order.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 48.5.39 order_submitter_t (named structure 34819)

Variable	Description
submitter	trading_code_t See trading_code_t sub structure.



#### 48.5.40 order_owner_t (named structure 34804)

Variable	Description
owner	trading_code_t See trading_code_t sub structure.

### 48.5.41 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.

#### 48.5.42 crossing_t (named structure 34820)

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

#### 48.5.43 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.

#### 48.5.44 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.

#### 48.5.45 short_sell_order_change_t (named structure 34830)

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

#### 48.5.46 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 three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

### 48.5.47 give_up_member_t

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

### 48.5.48 order_var_t

Variable	Description
mp_quantity_i	int64_t
	Shown quantity of the order.
premium_i	int32_t
	Premium - the price of the order. A combination of this field and the order_type_c field
	signify different types of orders.
	0 = Market order (order_type_c > 1)
	Any value = Fixed price order (order_type_c=1).
	Note that the price for a tailor made combination order can be positive, zero or negative.
block_n	uint32_t
	Block size. Possible values include:
	0 = Fill Or Kill order (time_validity_n = 0)
	1 = All other orders types.
time_validity_n	uint16_t
	This field is made up of two 8-bit parts - unit (most significant byte) and count (less
	significant byte).
	Unit = 0, Count = 0 (i.e. binary = 0): Order is an "Immediate" type order. Fill Or Kill when
	block_n = 0. Fill And Kill when block_n = 1.
	Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of
	the day.



Variable	Description
	<ul> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 (user order_type_c to determine Limit or Market-to-limit).</pre>
ex_client_s	char[10] Client - a free text field typically used to indicate to participants the ultimate client making the order.
customer_info_s	char[15] Customer information - a free text field typically used to indicate to the participants 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 include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> <i>Information</i> .
order_type_c	uint8_t


Variable	Description		
	Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer) 2 = Market order (premium_i = 0)		
	<ul> <li>3 = Market order (premium_i = 0)</li> <li>3 = Market-to-Limit order (premium_i = 0)</li> <li>17 = Best-Limit order (premium_i = 0, time_validity_n != 0)</li> <li>65 = Imbalance Limit order (premium_i = an integer).</li> <li>Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Centre Point, Centre Point Block and Sweep orders.</li> </ul>		
stop_condition_c	uint8_t Ignore. Currently not used.		
filler_2_s	char[2] Ignore. Used for byte alignment.		

# 48.5.49 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.

# 48.5.50 trading_code_t

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

## 48.5.51 originator_trading_code_t

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



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

### 48.5.52 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.

# 48.5.53 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 include:		
	0 = No value		
	1 = Initial trade report		
	2 = No initial trade report.		
filler_1_s	char[1]		
	Ignore. Used for byte alignment.		
extended_price_q	int64_i		
	This field may be set to either the reported trade price with up to four decimal places (the minimum value is 1000, indicating a value of 0.1000), or the special value indicating that it is to be ignored. If the 63 rd bit (highest bit) is set and the rest are zero, then this indicates that there is no extended price available.		

## 48.5.54 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



Variable	Description	Character Position	ASIC defined content
			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 (default).
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.



## 49 MQ392 Total Order for BSPs Query

### 49.1 Query Function

This query can only be used by Broker Service Providers (BSPs). It is used to retrieve a particular participant's orders. It has the same structure as the MQ8 query. However, BSPs are allowed to place a different participant's trading code in the whose_t struct, other than their own, thereby accessing the orders of that participant. ASX Trading Operations are responsible for configuring to which participant a BSP gains access.



### Note:

This query is a legacy query that does not return all order details. It only returns those fields that can be specified for an order using the legacy MO31 Enter Order transaction. To retrieve the full order details, use the MQ476 Own Orders for BSPs query.

### 49.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_tot_order_p_t	
Partitioned	True	
Segmented	False	
Answers	MA43	

#### 49.3 Answer Properties

Transaction Type	MA43
Struct Name	answer_tot_order_t

#### 49.4 Message Structure

#### 49.4.1 query_tot_order_p_t

Variable	Description			
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 392}.			
series	series_t To retrieve all a participants own orders in the system on the first query, zero-fill this series field and the order_index_u field. On subsequent queries users are to fill in the same two fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field cannot act as a filter. The query retrieves all orders for a particular participant.			
whose	whose_t Used to filter the results. See whose_t sub structure below.			
order_index_u	uint32_t			



Variable	Description
	Used as a coupling field for retrieving partitioned information. Use the series and this field to retrieve partitioned information.

### 49.4.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters in the query. It can be configured to specify a particular user's, participants, a user's client's or a participants client's orders.

Type of Order	Fields to be completed
User's Orders	country_id_s, ex_customer_s, user_id_s
Participants Orders	country_id_s, ex_customer_s
User's Orders for a Specific Client	country_id_s, ex_customer_s, user_id_s ex_client_s
Participants Orders for a Specific Client	country_id_s, ex_customer_s and ex_client_s



#### Note:

Fields that are omitted should be filled with NULLs and are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t BSPs indicate here which participant they are interested in. The country_id_s and the ex_customer_s fields are mandatory. See trading_code_t sub structure below.
ex_client_s	char[10] When placing an order, this field is a free text and is typically used to indicate to the participant the ultimate client making the order. In this structure it can be used as a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 49.4.3 trading_code_t

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



Variable	Description
	The unique identifier of the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

## 49.5 Answer Structure

After a successful MQ392 transaction, a list of the participant's orders in the order book is returned to the sender.

49.5.1	answer	tot	ord	er t	t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 43}.
series	series_t The value in this field, along with the order_index_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_index_u	uint32_t The value in this field, along with the series value, should be used in the next query.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_tot_order_item_t[300] See answer_tot_order_item_t sub structure below.

## 49.5.2 answer_tot_order_item_t

Variable	Description
order_number_u	quad_word The order identifier for the order in this item.
sequence_number_u	uint32_t Ignore. Currently not used.
ob_position_u	uint32_t Indicates the order book position, 1 being the highest priority.
combo_mark_c	uint8_t Ignore. Not used in this query.
filler_3_s	char[3] Ignore. Used for byte alignment.
party	party_t See party_t sub structure.
order	order_t Detailed information on the order. See order_t sub structure below.
total_volume_i	int64_t The total quantity of the order i.e. both the hidden and shown (display_quantity_i) volume.



Variable	Description
	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.
display_quantity_i	<ul> <li>int64_t</li> <li>This value should only be considered if the order is an iceberg order. It indicates the original displayed quantity of the order and is therefore the value to which the iceberg order will use to re-populate its display_quantity_i field once all its current shown quantity has been traded.</li> <li>ASX Trade will automatically populate this field when the total_volume_i field != 0.</li> </ul>
orig_shown_quantity_i	int64_t The shown quantity of the order when it was originally entered.
orig_total_volume_i	int64_t The total_volume_i value of the order when it was originally entered.
timestamp_in	time_spec_t The time when the order was entered <b>or</b> when it was last modified. See time_spec_t sub structure.
timestamp_created	time_spec_t The time when the order was originally entered. See time_spec_t sub structure.

# 49.5.3 party_t

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

# 49.5.4 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



Variable	Description
	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 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. 31 indicates MO31).
change_reason_c	<ul> <li>uint8_t</li> <li>Indicates the reason for the change. Possible values include: <ol> <li>Order deleted</li> <li>Trade</li> <li>Order inactivated</li> <li>Order added</li> <li>Order added</li> <li>Order deleted by trade system</li> <li>Order deleted by trade system</li> <li>Order deleted by trade system</li> <li>Order deleted by proxy</li> <li>Hidden volume order recalculated</li> <li>Trade system deleted day order</li> <li>I = Inactivated due to no shown quantity</li> <li>I = lnactivated due to Purge</li> <li>I = lnactivated due to Purge</li> <li>I = lnactivated due to Purge</li> <li>I = lnactivated due to Price away from the market</li> <li>Order reload at normal system start</li> <li>Order reload at intraday Market Place restart</li> <li>Cancelled After Auction</li> <li>Convert undisclosed order to normal order - for active orders falling below the minimum order value due to Market Maker protection delta limit reached or exceeded</li> <li>Quote deleted due to Market Maker protection delta limit reached or exceeded</li> <li>Quote deleted due to Market Maker protection delta limit reached or exceeded</li> <li>Market-to-Limit Sweep order converted to Limit order</li> <li>Centre Point Block or Sweep order has traded below its MAQ and MAQ is reset to zero</li> </ol></li></ul>
	char[1] Ignore. Used for byte alignment.



## 49.5.5 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.
premium_i	<pre>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 (order_type_c &gt; 1) Any value = Fixed price order (order_type_c=1). Note that the price for a tailor made combination order can be positive, zero or negative.</pre>
block_n	uint32_t Block size. Possible values include: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 (user order_type_c to determine Limit or Market-to-limit)</pre>



Variable	Description
	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 participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> Information.
order_type_c	<pre>uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer) 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.</pre>
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 49.5.6 trading_code_t/ex_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 three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user. Users can retrieve their own identifier using the omniapi_get_info_ex() function.



## 49.5.7 give_up_member_t

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



## 50 MQ393 Total Inactive Order for BSPs Query

#### 50.1 Query Function

This query can only be used by Broker Service Providers (BSPs). It is used to retrieve a particular participant's inactive orders. It has the same structure as the MQ9 query. However, BSPs are allowed to place a different participant's trading code in the whose_t struct, other than their own, thereby accessing the inactive orders of that participant. ASX Trading Operations are responsible for configuring to which participants a BSP gains access.



### Note:

This query is a legacy query that does not return all order details. It only returns those fields that can be specified for an order using the legacy MO31 and MO98 transactions. To retrieve the full order details, use the MQ476 Own Orders for BSPs query, which can also be used to query for inactive orders.

### 50.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	query_tot_order_p_t
Partitioned	True
Segmented	False
Answers	MA44

#### 50.3 Answer Properties

Transaction Type	MA44
Struct Name	answer_tot_order_t

#### 50.4 Message Structure

#### 50.4.1 query_tot_order_p_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 393}.
series	series_t To retrieve all a participant's inactive orders in the system on the first query zero-fill this series field and the order_index_u field. On subsequent queries users are to fill in the same two fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled. This field cannot act as a filter. The query retrieves all inactive orders for a particular participant.
whose	whose_t Used to filter the results. See whose_t sub structure below.
order_index_u	uint32_t



Variable	Description
	Used as a coupling field for retrieving partitioned information. Use the series and this field to retrieve partitioned information.

### 50.4.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters in the query. It can be configured to specify a particular user's, participant's, user's client's or participant's client's inactive orders.

Type of order	Fields to be Completed
User's orders	country_id_s, ex_customer_s, user_id_s
Participant's orders	country_id_s, ex_customer_s
User's orders for a specific client	country_id_s, ex_customer_s, user_id_s ex_client_s
Participant's orders for a specific client	country_id_s, ex_customer_s and ex_client_s



#### Note:

Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t BSPs indicate here which participant they are interested in. The country_id_s and the
	ex_customer_s fields are mandatory. See trading_code_t sub structure below.
ex_client_s	char[10] When placing an order this field is a free text and is typically used to indicate to participants the ultimate client making the order. In this structure it can be used as a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

### 50.4.3 trading_code_t

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



Variable	Description
	The unique identifier of the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.

### 50.5 Answer Structure

After a successful MQ393 transaction, a list of the participant's inactive orders in the order book is returned to the sender.

50.5.1 answer_	tot_	order_	t
----------------	------	--------	---

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 44}.
series	series_t The value in this field, along with the order_index_u value, should be used in the next query. A zero-filled structure indicates this is the last answer.
order_index_u	uint32_t The value in this field, along with the series value, should be used in the next query.
items_n	uint16_t The number of items held in the array.
filler_2_s	char[2] Ignore. Used for byte alignment.
item	answer_tot_order_item_t[300] An array containing the order details. See answer_tot_order_item_t sub structure below.

## 50.5.2 answer_tot_order_item_t

Variable	Description
order_number_u	quad_word The order identifier for the order in this item.
sequence_number_u	uint32_t Ignore. Currently not used.
ob_position_u	uint32_t Indicates the order book position, 1 being the highest priority.
combo_mark_c	uint8_t Ignore, not used in this query.
filler_3_s	char[3] Ignore. Used for byte alignment.
party	party_t See party_t sub structure.
order	order_t Detailed information on the order. See order_t sub structure below.
total_volume_i	int64_t



Variable	Description
	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
	lceberg order: total_volume_i = display_quantity_i != 0 lceberg order: total_volume_i > display_quantity_i != 0.
display_quantity_i	<ul> <li>int64_t</li> <li>This value should only be considered if the order is an iceberg order. It indicates the original displayed quantity of the order and is therefore the value to which the iceberg order will use to re-populate its display_quantity_i field once all its current shown quantity has been traded.</li> <li>ASX Trade will automatically populate this field when the total_volume_i field != 0.</li> </ul>
orig_shown_quantity_i	int64_t The shown quantity of the order when it was originally entered.
orig_total_volume_i	int64_t The total_volume_i value of the order when it was originally entered.
timestamp_in	time_spec_t The time when the order was entered when it was last modified. See time_spec_t sub structure.
timestamp_created	time_spec_t The time when the order was originally entered. See time_spec_t sub structure.

# 50.5.3 party_t

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

# 50.5.4 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.



Variable	Description
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 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. 98 indicates
	MO98).
change_reason_c	uint8_t
	Indicates the reason for the change. Possible values include:
	1 = Order deleted
	3 = Deal
	4 = Order inactivated
	5 = Order amended
	6 = Order added
	8 = Order price changed
	9 = Order deleted by trade system
	10 - Oldel deleted by ploxy 13 - Hidden volume order recalculated
	19 - Trade system deleted day order
	21 = Inactivated by system due to Instrument Session change
	22 = Deleted by system due to instrument session enable
	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 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 - Sween order reloaded without MAO and mid-tick attribute
filler 1 c	
IIIIer_1_S	Cildi[1]



## 50.5.5 order_var_t

Variable	Description
mp_quantity_i	int64_t Shown quantity of the order.
premium_i	<pre>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 (order_type_c &gt; 1) Any value = Fixed price order (order_type_c=1). Note that the price for a tailor made combination order can be positive, zero or negative.</pre>
block_n	uint32_ Block size. Possible values: 0 = Fill Or Kill order (time_validity_n = 0) 1 = All other orders types.
time_validity_n	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 (user order_type_c to determine Limit or Market-to-limit)</pre>



Variable	Description
	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 participants the ultimate client making the order.
customer_info_s	char[15] Customer information – a free text field typically used to indicate to participants their own order identifier.
open_close_req_c	uint8_t Ignore. Currently not used.
bid_or_ask_c	uint8_t Bid or Ask. Possible values include: 1 = Buy 2 = Sell.
ext_t_state_c	uint8_t Trade report code. Possible values include: 0 = Ignore, not relevant. For other values see <i>Trade Report Types</i> in <i>ASX Trade Introduction and Business</i> <i>Information</i> .
order_type_c	<pre>uint8_t Order Type – a combination of this field and the premium_i field signify different types of orders. Possible values include: 1 = Limit price order (premium_i = an integer) 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.</pre>
stop_condition_c	uint8_t Ignore. Currently not used.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 50.5.6 trading_code_t/ex_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 three digit number whereas information venders have an alphanumeric identifier. The combination of the country_id_s field and this field uniquely define a trading participant.
user_id_s	char[5] The unique identifier of the user who placed the order. Users can retrieve their own identifier using the omniapi_get_info_ex() function.



## 50.5.7 give_up_member_t

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

# 50.5.8 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.



## 51 MQ476 Own Orders Query for BSPs

### 51.1 Query Function

This query can only be used by Broker Service Providers (BSPs). It is used to retrieve a particular participant's orders. It has the same structure as the MQ92 query. However, BSPs are allowed to place a different participant's trading code in the whose_t struct, other than their own, thereby accessing the orders of that participant. ASX Trading Operations are responsible for configuring to which participants a BSP gains access. The query can be used to query for both active and central inactive orders.

### 51.2 Query Properties

Function Call	omniapi_query_ex
Facility	EPO
Struct Name	query_order_private_filter_p_t
Partitioned	True
Segmented	True
Answers	MA476

#### 51.3 Answer Properties

Transaction Type	MA476
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the
	sequence of possible structs are described below.

#### 51.4 Message Structure

### 51.4.1 query_order_private_filter_p_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'M', 'Q', 476}.
series	series_t To retrieve all own orders, on the first query, zero-fill this series field, the order_number_u field and bid_or_ask_c. On subsequent queries users are to fill in the series and order_number_u fields with values received from the answer. The final response is indicated by the series field in the answer being zero-filled.
search_series	series_t Acts as a filter for this query. Fill in country and market code to retrieve information on that particular market. Fill in country, market and instrument group code to retrieve information on that particular instrument type. Fill in country, market, instrument group and commodity code to retrieve information on that particular instrument class. Or provide a full series definition to query for a particular instrument series.
whose	Whose_t Used to filter the results. See whose_t sub structure below.



Variable	Description
order_number_u	quad_word Use this field to retrieve information that requires more than one response. On the first query set this field to zero, then on subsequent queries, used the value in the order_number_u field in the answer. A zero value in the order_number_u field indicates the last response.
order_filter_i	<ul> <li>int32_t</li> <li>Used to filter the results on the types of orders to return.</li> <li>Acts as a bit map that defines which orders will be returned by the query. Several bits can be combined to return all orders matching the selected bit filters. To query for all types of orders, either set the field empty or set all bits. Possible values:</li> <li>1 = All active orders (excluding Centre Point orders)</li> <li>2 = All active quotes entered with quoting transactions MO36 and MO37</li> <li>4 = All active Centre Point, Centre Point Block and dual-posted Sweep orders</li> <li>8 = All stop orders (not used by ASX)</li> <li>16 = All central inactive orders and quotes (including central inactive Centre Point and Centre Point Block orders).</li> </ul>
bid_or_ask_c	uint8_t Always set to zero in this query.
filler_3_s	char[3] Ignore. Used for byte alignment.

# 51.4.2 whose_t

The strings in trading_code_t and ex_client_s are used as search parameters in the query. It can be configured to specify a participant's user, clients, user's Client's or Client's orders.

Type of Order	Fields to be Completed
My Orders	country_id_s, ex_customer_s, user_id_s
Our Orders	country_id_s, ex_customer_s
My Orders for a Specific Client	country_id_s, ex_customer_s, user_id_s ex_client_s
Our Orders for a Specific Client	country_id_s, ex_customer_s and ex_client_s



### Note:

Fields that are omitted should be filled with NULLs, they are not to be space padded. Furthermore, unlike the use of the whose_t structure in the MO4 transaction, the ex_client_s field **cannot** contain wildcards.

Variable	Description
trading_code	trading_code_t See trading_code_t sub structure below.
ex_client_s	char[10] When placing an order this field is a free text, and is typically used to indicate to participants the ultimate client for this side of the trade. In this structure it can be used as



Variable	Description
	a filter so that the answer contains only those orders that have an ex_client_s matching the given string.
filler_2_s	char[2] Ignore. Used for byte alignment.

#### 51.5 Answer Structure

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

- answer_order_hdr_t
- one or more sequences of:
  - item_hdr_t
  - one or more sequences of:
    - sub_item_hdr_t
    - a choice of:
      - basic_order_book_order_t (named structure 34817)
      - reserve_order_t (named structure 34812)
      - 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)
      - ranking_time_t (named structure 34949)
      - crossing_t (named structure 34820)
      - regulatory_t (named structure 34821)
      - short_sell_order_t (named structure 34829).

#### 51.5.1 answer_order_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'M', 'A', 476}.
next_series	series_t The value in this field should be used in the next query. A zero-filled structure indicates this is the last answer.
next_order_number_u	quad_word The value in this field, along with the series, should be used in the next query.
bid_or_ask_c	uint8_t Ignore. Not used for this query. The next query should always be sent with bid_or_ask_c set to zero.
filler_3_s	char[3] Ignore. Used for byte alignment.
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.

# 51.5.2 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.	

## 51.5.3 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.

# 51.5.4 basic_order_book_order_t (named struct number 34817)

Variable	Description
series	series_t
	The series for which the order exists.
owner	trading_code_t
	Indicates the user who originally placed the order.
	See trading_code_t sub structure below.
give_up_member	give_up_member_t
	The clearing identifier used for the order.
	See give_up_member_t sub structure below.
order_number_u	quad_word
	The order identifier for the order in this item.
timestamp_in	time_spec_t
	The time when the order was entered OR when it was last modified.
	See time_spec_t sub structure.
timestamp_created	time_spec_t
	The time when the order was first entered.
	See time_spec_t sub structure.
sequence_number_u	uint32_t
	Ignore. Currently not used.
ob_position_u	uint32_t
	Indicates the order book position, 1 being the highest priority.
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).



Variable	Description
	The price for a tailor made combination 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.
original_quantity_i	int64_t The quantity that the order was originally entered with. For iceberg orders this is the original total quantity of the order. The original 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	<ul> <li>uint16_t</li> <li>This field is made up of two 8-bit parts - unit (most significant byte) and count (less significant byte).</li> <li>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.</li> <li>Unit = 1, Count = 0 (i.e. binary 1 0000 0000, hex 100, dec 256): Order is valid for the rest of the day.</li> <li>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.</li> <li>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.</li> <li>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.</li> <li>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.</li> </ul>
exch_order_type_n	<pre>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 premium_i = an integer, order_type_c = 65 = Imbalance Limit order) Can also be combined with other exchange specific order types/attributes) outlined below. 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)</pre>



Variable	Description
	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).
transaction_number_n	uint16_t The transaction number that was used to originally enter this order (e.g. 1 indicates MO1).
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.
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.
order_type_c	<ul> <li>uint8_t</li> <li>Order type – a combination of this field and the premium_i field signifies different types of orders. Possible values:</li> <li>1 = Limit price order (premium_i = an integer)</li> <li>2 = Market order (premium_i = 0)</li> <li>3 = Market-to-Limit order (premium_i = 0)</li> <li>17 = Best-Limit order (premium_i = 0 and time_validity_n !=0</li> <li>65 = Imbalance Limit order (premium_i = an integer).</li> <li>Used in conjunction with values in the field exch_order_type_n to determine Undisclosed, Sweep, Centre Point and Centre Point Block orders.</li> </ul>
bid_or_ask_c	uint8_t Bid or Ask. Possible values: 1 = Buy 2 = Sell.
change_reason_c	<ul> <li>uint8_t</li> <li>Indicates the reason for the change. Possible values include:</li> <li>1 = Order deleted</li> <li>3 = Trade</li> <li>4 = Order inactivated</li> <li>5 = Order amended</li> <li>6 = Order added</li> <li>8 = Order price changed</li> <li>9 = Order deleted by trade system</li> <li>10 = Order deleted by proxy</li> <li>13 = Hidden volume order recalculated</li> <li>19 = Trade system deleted day order</li> <li>21 = Inactivated by system due to Instrument Session change</li> <li>22 = Deleted by system due to no shown quantity</li> <li>23 = Inactivated due to Purge</li> <li>24 = Inactivated due to Expiry</li> <li>27 = Inactivated due to Price away from the market</li> </ul>



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 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.

# 51.5.5 trading_code_t

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

## 51.5.6 give_up_member_t

Variable	Description
country_id_s	char[2]
	For ASX Trade this is always set to AU , indicating the Australian exchange.
ex_customer_s	char[5] This is a unique clearing identifier. Possible values for a 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.

# 51.5.7 time_spec_t

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



Variable	Description
tv_nsec	int32_t
	Elapsed time in nanoseconds since the seconds in tv_sec.

# 51.5.8 reserve_order_t (named structure 34812)

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

## 51.5.9 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.

## 51.5.10 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.
mid_tick_c	<ul> <li>uint8_t</li> <li>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).</li> <li>Specifies whether a Limit Sweep order is eligible for passive execution in Centre Point at a half-tick above the limit price.</li> <li>0 = not defined</li> <li>1 = mid-tick attribute set on</li> <li>2 = mid-tick attribute set off</li> </ul>
	<ul> <li>3 = allowed for permitted prices in addition to mid-point ('dark limit' order)</li> <li>4 = allowed for permitted prices in addition to mid-point ('dark limit' order), with mid-tick attribute set on.</li> <li>5 = Any Price Block order</li> <li>6 = Any Price Block order with mid-tick attribute set to on.</li> <li>Limit Sweep orders fully integrate the liquidity in ASX Centre Point and ASX TradeMatch and will interact with both 'mid-point only' and 'dark limit' Centre Point and Centre Point Block orders and Any Price Block orders. The only allowed mid_tick_c values for Limit Sweep orders are 0, 1 or 2.</li> </ul>
preference_only_c	uint8_t



Variable	Description
	Specifies whether a Centre Point Order or a Centre Point Block order is a Preference and
	Kill order or not. For Limit Sweep orders, only '0' or '2' are valid values.
	0 = not defined or 'no', order is not a Preference and Kill order.
	1 = 'yes', order is a Preference and Kill order. Time validity must be set to Fill and Kill or Fill
	or Kill for this option.
	2 = 'no', order is not a Preference and Kill order.
	This variable cannot be amended.
single_fill_minimur	n_quantit uint8_t
y_c	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.

## 51.5.11 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 1 = central inactive order 2 = active order.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 51.5.12 order_submitter_t (named structure 34819)

This structure is included when an order has been entered by ASX Trading Operations on behalf of a participant, showing the user who entered the order.

Variable	Description
submitter	trading_code_t See trading_code_t sub structure above.

## 51.5.13 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 above.

## 51.5.14 crossing_t (named structure 34820)

Variable	Description
crossing_key_i	int32_t



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

## 51.5.15 regulatory_t (named structure 34821)

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

## 51.5.16 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	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 (default).
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.

## 51.5.17 short_sell_order_t (named structure 34829)

Variable	Description
short_sell_quantity_i	<pre>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.</pre>



# 52 UQ1 Partition Query

# 52.1 Query Function

This query will return all partition information.

## 52.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_partition_t	
Partitioned	False	
Segmented	True	
Answers	UA1	

## 52.3 Answer Properties

Transaction Type	UA1
Struct Name	answer_partition_t

## 52.4 Message Structure

## 52.4.1 query_partition_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'U', 'Q', 1}.
series	series_t Set to zero.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
filler_2_s	char[2] Ignore. Used for byte alignment.

## 52.5 Answer Structure

## 52.5.1 answer_partition_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 1}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.



Variable	Description	
items_n	uint16_t The number of items held in the array.	
item	answer_partition_item_t[100] See <i>answer_partition_item_t</i> sub structure below.	

## 52.5.2 answer_partition_item_t

Variable	Description
server_partition	server_partition_t
	See server_partition_t sub structure below.

## 52.5.3 server_partition_t

Variable	Description
server_name_s	char[20] Name of the server.
transaction_type_high	transaction_type_t Defines the high end in a range of transactions in one partition. This field and the transaction_type_low field define a set permissible transaction for this partition.
transaction_type_low	transaction_type_t Defines a low end in a range of transactions in one partition. This field and the transaction_type_high field define a set permissible transaction for this partition.
series_fields_used	series_t Indicates the fields that are used in the partition_low and partition_high fields. Possible values include: 0 = field is not used 1 = field is used.
partition_low	series_t Defines the low end in a range of consecutive series in one partition. This field and the partition_high field define a range of series that exist in this partition.
partition_high	series_t Defines the high end in a range of consecutive series in one partition. This field and the partition_low field define a range of series that exist in this partition.
event_type_i	int32_t Ignore, not used.



# 53 UQ9 B17 Signals Sent Query

# 53.1 Query Function

This query is used to retrieve all Information (BI7) signals that were sent on a particular day.

## 53.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_bi7_signals_sent_t	
Partitioned	False	
Segmented	True	
Answers	UA9	

## 53.3 Answer Properties

Transaction Type	UA9
Struct Name	answer_bi7_signals_sent_t

### 53.4 Message Structure

## 53.4.1 query_bi7_signals_sent_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'U', 'Q', 9}.
search_series	series_t Set to zero.
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
business_date_s	char[8] Date in YYYYMMDD format.
seq_num_srm_n	uint16_t Ignore. Set to zero.

## 53.5 Answer Structure

### 53.5.1 answer_bi7_signals_sent_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 9}.
segment_number_n	uint16_t



Variable	Description
	Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_bi7_signals_sent_item_t[1000] See answer_bi7_signals_sent_item_t sub structure below

# 53.5.2 answer_bi7_signals_sent_item_t

Variable	Description
series	series_t
	The market or series for which the original broadcast was sent.
info_type_i	int32_t
	The type of information that is ready. Possible values include:
	100 = Settlement information
	13 = All securities closed
	1 = Trade information.
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
	Ignore. Not used.



## 54 UQ12 Business Date

## 54.1 Query Function

The purpose of this query is to get the current business date, the UTC and time.

Note that this information is not for time synchronisation purposes. For synchronisation purposes use Network Time Protocol (NTP).

The answer also contains the exchanges TZ-variable and the current offset between UTC and the local time specified in the TZ-variable.

## 54.2 Query Properties

omniapi_query_ex	
EP1	
query_business_date_t	
False	
False	
UA12	
	omniapi_query_ex EP1 query_business_date_t False False UA12

### 54.3 Answer Properties

Transaction Type	UA12
Struct Name	answer_business_date_t

### 54.4 Message Structure

### 54.4.1 query_business_date_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'II' 'O' 12}

### 54.5 Answer Structure

## 54.5.1 answer_business_date_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 12}.
omex_version_s	char[16] This is the current OI version running on the system.
business_date_s	char[8] Business date. Format: YYYYMMDD.
utc_date_s	char[8]



Variable	Description
	UTC date.
	Format: YYYYMMDD.
utc_time_s	char[6]
	UTC time.
	Format: HHMMSS.
tz_variable_s	char[40]
	The TZ environment variable for the exchange (POSIX standard).
	e.g. MET-1MET_DST-2,M3.5.0/2,M10.5.0/3
filler_2_s	char[2]
	Ignore. Used for byte alignment.
utc_offset_i	int32_t
	Current offset in minutes between UTC and the local time specified in the TZ-variable.



## 55 UQ14 BI81 Broadcasts Sent Query

## 55.1 Query Function

This query retrieves BI81 broadcasts that have been sent on the current or previous business day.

### 55.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_bi81_broadcasts_sent_t	
Partitioned	False	
Segmented	True	
Answers	UA14	

### 55.3 Answer Properties

Function Call	UA14
Struct Name	The message complies with the VIM concept and has no topmost struct. Instead, the
	sequence of possible structs are described below.

### 55.4 Message Structure

### 55.4.1 query_bi81_broadcasts_sent_ta

The following fields can be filled with zeros as wildcards to retrieve all messages sent for a certain date:

- message_information_type_c
- message_priority_c
- search_series.

Variable	Description
transaction_type	transaction_type_t
	Set the structure to the following: {'U', 'Q', 14}.
series	series_t
	Set all fields to zero to search for all BI81 broadcasts.
	For all BI81 broadcasts relating to a certain underlying, fill in commodity_n. To search for
	BI81 broadcasts for a certain market, fill in country_c and market_c.
segment_number_n	uint16_t
	Ignore. Set to 1.
message_information_typ	e_c uint8_t
	Used as a filter to retrieve the broadcasts that contains the relevant types of messages.
	Possible values include:
	0 = wildcard (all types)
	1 = Company Announcement
	2 = Market Message
	3 = Static Line
	4 = Notice Received.


Variable	Description
message_priority_c	uint8_t Used as a filter to retrieve the broadcasts that contain the relevant types of priority. Possible values: 0 = wildcard (all priorities) 1 = low priority 2 = medium priority 3 = high priority 4 = critical priority.
date_s	char[8] Set to the current business date or the previous trading day's date in YYYYMMDD format. <b>Note</b> : If date is zero-filled, today's business date is assumed.
from_sequence_number_u	uint32_t First sequence number in the range identifying the broadcasts the user wants to retrieve. The sequence numbers for the BI81 start at 1 each day.
to_sequence_number_u	uint32_t Last sequence number in the range identifying the broadcasts the user wants to retrieve. The sequence numbers for the BI81 start at 1 each day. A zero indicates all BI81s from the value stated in the from_sequence_number_u field.
search_series	series_t Set all fields to zero to search for all series. To search for a specific market, fill in country_c and market_c. Fill in the fill series definition to search for a specific series.
update_status_note_c	uint8_t Ignore. Set to zero.
filler_3_s	char[3] Ignore. Used for byte alignment.

### 55.5 Answer Structure

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

- answer_segment_hdr_t
  - one or more sequences of:
  - item_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).



# 55.5.1 answer_segment_hdr_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 14}.
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 Indicates the segment for this particular reply. A value of zero means that there are no more segments.
filler_2_s	char[2] Ignore. Used for byte alignment.

# 55.5.2 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.

## 55.5.3 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.

### 55.5.4 message_core_info_t (named structure 35001)

Variable	Description
sequence_number_u	uint32_t
	Sequence number assigned to the broadcast.
message_information_type_	c uint8_t
	Identifies the kind of message sent. Possible values include:
	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]
	Date of the broadcast.
	Format: YYYYMMDD.



Variable	Description
hhmmss_s	char[6]
	Time of the broadcast.
	Format: HHMMSS.
message_priority_c	uint8_t
	Possible values include:
	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 use.
filler_3_s	char[3]
	Ignore. Used for byte alignment.

# 55.5.5 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 message_information_item_t sub structure below.

# 55.5.6 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.

# 55.5.7 destination_item_t (named structure 35003)

Variable	Description
series	series_t Indicates the market, underlying or series to which the original broadcast related. Note 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 include: 1 = Market level



Variable	Description
	2 = Underlying level 3 = Series level.
filler_3_s	char[3] Ignore. Used for byte alignment.

## 55.5.8 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.



## 56 UQ15 Instrument Status

### 56.1 Query Function

This query will return the session state for a market, instrument type, underlying instrument class, series or for all instrument levels.

#### 56.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EP1	
Struct Name	query_instrument_status_t	
Partitioned	False	
Segmented	True	
Answers	UA15	

#### 56.3 Answer Properties

Transaction Type	UA15
Struct Name	answer_instrument_status_t

#### 56.4 Message Structure

#### 56.4.1 query_instrument_status_t

The query searches using the parameters set in series and the state_level_e.

The session state is updated by the BI41 and BI741 broadcasts. The UQ15 returns information in the same format as the BI41.

More information about the trading session handling is found in *Trading Session, Instrument Session and Active Session States* in ASX Trade Introduction and Business Information.

Variable	Description	
transaction_type	transaction_type_t Set the structure to the follo	owing: {'U', 'Q', 15}.
series	series_t Should be completed accord Any of the fields filled with z the series are filled with zer instrument classes, series ar	ding to the table below. zero are regarded as a wildcard for that field. If all the fields in os, all session states for all markets, instrument types, nd underlyings will be returned.
	What to identify	Fill in these Fields
	Market	country_c market_c
	Instrument Type	country_c market_c instrument_group_c



Variable	Description		
	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	
	Underlying, linked underlying	commodity_n	
segment_number_n	uint16_t Indicates which segment the user structure has a segment number f	wishes to receive, typically starting at one. The reply ield too, which indicates which segment was returned.	
state_level_e	uint16_t Can be used to filter the results by 0 = All levels 1 = Market 2 = Instrument Type 3 = Instrument Class 4 = Instrument Series 5 = Underlying, Linked Underlying.	the level the session state is applied to.	

### 56.5 Answer Structure

## 56.5.1 answer_instrument_status_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 15}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_instrument_status_item_t[1000] See answer_instrument_status_item_t sub structure below.

## 56.5.2 answer_instrument_status_item_t

Variable	Description
series	series_t



Variable	Description		
	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 and at what levels.		
	What to identify	Fill in these Fields	
	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	
	Underlying, linked underlying	commodity_n	
state_number_n	uint16_t The binary representation of th Available values can be fetched	ne session state. d through the DQ29 Trading State query.	
state_level_e	uint16_t Indicates the level that a state 1 = Market 2 = Instrument Type 3 = Instrument Cass 4 = Instrument Series 5 = Underlying, Linked Underly	applies to. Possible values: ring.	



### 57 UQ20 BI73 Signals Sent

### 57.1 Query Function

This query is used to determine whether any BI73 (Undo Information Ready) broadcasts have been sent previously on a certain day. If no such broadcast was sent on that particular day for the given market then an OMNIAPI_NO_SUCH_ID (-17) error will be returned.

### 57.2 Query Properties

Function Call	omniapi_query_ex	
Facility	EPO	
Struct Name	query_bi73_signals_sent_t	
Segmented	True	
Partitioned	False	
Answers	UA20	

### 57.3 Answer Properties

Transaction Type	UA20
Struct Name	answer_bi73_signals_sent_t

### 57.4 Message Structure

### 57.4.1 query_bi73_signals_sent_t

Variable	Description
transaction_type	transaction_type_t Set the structure to the following: {'U', 'Q', 20}.
search_series	series_t Fill in country_c and market_c to retrieve information on that particular market (mandatory fields).
segment_number_n	uint16_t Indicates which segment the user wishes to receive, typically starting at one. The reply structure has a segment number field too, which indicates which segment was returned.
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.
seq_num_srm_n	uint16_t Ignore. Set to zero.



### 57.5 Answer Structure

# 57.5.1 answer_bi73_signals_sent_t

Variable	Description
transaction_type	transaction_type_t Contains the following: {'U', 'A', 20}.
segment_number_n	uint16_t Indicates the segment for this particular reply. A value of zero means that there are no more segments.
items_n	uint16_t The number of items held in the array.
item	answer_bi73_signals_sent_item_t[1000] See answer_bi73_signals_sent_item_t sub structure below.

# 57.5.2 answer_bi73_signals_sent_item_t

Variable	Description
series	series_t Used to identify a specific market, instrument type, instrument class, series or underlying.
info_type_i	int32_t The type of information that is undone. Possible values include: 100 = Settlement information 13 = All securities closed 1 = Trade information.
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.
seq_num_srm_n	uint16_t A unique sequence number.



#### 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.

 $\mathsf{ASX}^{*}$