ASX Trade FIX Order Entry Conformance Process

Version 1.0 January 2026

Contents

1. Introduction

The ASX application conformance test process is designed to help protect market participants from any erroneous application behaviour that may be disruptive to the market due to non-conformance.

The conformance process applies to any application that connects to the ASX Trade Production environment for FIX Order Entry services.

All customer applications must pass the ASX Conformance process prior to accessing the Production market.

Except for certain mandatory functions, customers are required to complete only the conformance test scenarios which relate to the functionality they will use in the Production environment. All other test scenarios may be omitted.

1.1. Purpose

The purpose of this document is to provide an overview of the ASX Trade FIX Order Entry Conformance test and set expectations regarding conduct and expected results.

1.2. Readership

This document outlines the test scenarios customers need to pass to certify their order entry software applications.

Technical staff within ASX Trade participants and information vendors and other market participants that connect directly to the ASX Trade Platform are expected to read this document and understand the requirements of this process.

1.3. Document history

Issue	Date	Description
1.0	January 2026	First release of document

1.4. Enquiries

Please contact CTS@asx.com.au or your Technical Account Manager if you have any questions relating to this document.

2. ASX Conformance Process

The ASX conformance process provides guidelines for customers to ensure their software conforms to ASX Trade operating rules and technical specifications.

ASX expects customers to become conversant with the ASX development platform and develop their applications to a state of readiness for conformance testing. At this point they can contact ASX to schedule a date and time to undertake the conformance test. ASX requires customers to have finalised their software, meaning that said software is in a production-ready state, before booking the test.

The conformance process is as follows.

- Customers must submit their conformance checklist to CTS and clearly indicate the following:
  - The application name and version
  - The usernames being utilized to perform the scenarios
  - The test scenarios their application will not complete along with the reason
  - Start and end time of their self-test
- Customers must then advise CTS once they have completed performing their self-test scenarios; CTS will proceed to validate the outcome of the self-test. Once the self-test scenarios are validated as complete by CTS, customers can go on to perform the assisted test scenarios live on a call by a member of the CTS team. These assisted test slots are of 30 minutes duration, where if not completed inside this window customers will need to book an additional slot.
- Upon completion of the assisted testing, CTS will then validate the results of the overall conformance test.

2.1. When to Repeat Conformance Testing

While it is encouraged to perform regular testing, customers are required to perform software conformance with the ASX when:

- Software is modified in any way that may affect or directly impact ASX connectivity or messaging
- Additional ASX facing functionality is added to an already conformed application
- Software is recompiled for a new operating system
- ASX upgrades or changes its production environment and deems the change mandatory
- During extended periods of absence when the software is not connected to ASX
- Erroneous or disruptive behaviour is identified in the Production Environment
- Upon specific request from the ASX

ASX software conformance testing is optional but recommended when:

- Customer software is recompiled on the same operating system with minor changes, or new builds which have no assumed impact to ASX connectivity or messaging
- Independent Software Vendor (ISV) applications have passed conformance testing and the same software and version is to be used by new or existing customers of the ISV on new customer infrastructure

2.2. Non-Compliant software

⚠ **Warning:**

ASX reserves the right to block access to the production system by non-compliant software. ASX will require successful completion of a conformance test prior to re-connection.

2.3. Conformance Testing

To successfully complete Application Conformance Testing the customer’s software application needs to have established a connection to the ASX Trade Testing Environment (CDE/CDE+). These platforms simulate the ASX production environment and are designed to give a “Production-like” platform to develop and test against.

2.4. Prior to Conformance

Customers should ensure they have read and understand the specification documents pertaining to the interfaces they are developing against.

Customers should then discuss the intended functionality of their applications with their TAMs to ensure all steps were followed and the process is fully understood.

The CTS team will be available to assist customers in completing their tests, if required.

Customers must submit the application conformance checklist as provided by the CTS team with all functions not supported by the software clearly marked.

The checklist must be sent to CTS@asx.com.au prior to the test to validate the successfully attempted test scenarios.

2.5. Considerations for Software Vendors

All Participants are expected to create and manage orders/trade reports under their own Participant ID (PID) as part of the FIX Order Entry conformance process.

2.6. Expectations during Conformance

- Customer must maintain a stable connection
- Customers should refer to the ASX Operating Rules for guidelines on acceptable market behaviour
    - Adherence to these guidelines is required throughout all customer application conformance testing
    - Failure to comply with these guidelines will result in failure of the conformance test

2.7. Categories

Mandatory – must be attempted and passed.

Supported Functionality – this case is conditionally mandatory if you intend to support this functionality in Production.

Not Supported – on the checklist, Supported Functionality scenarios can be set to *Not Supported* under the following conditions:

- When the functionality will not be used in Production
- The conformance checklist clearly states the reason it is Not Supported

3. Self-testing Conformance Scenarios

The customer is responsible for completing this section of the conformance test. All activities performed during this segment of the conformance test are logged. On completion of this segment of the conformance test, ASX will review logs to ensure that applicable test scenarios were successfully completed by the customer application.

3.1. Session

3.1.1. Logon

Category	Mandatory	
Description	The application must perform a standard Logon. The application must maintain connectivity and receive a successful connection acknowledgement. The application must pass this test to progress to the next phase of functionality testing.	
Requirements	It is expected that the Participant will have confirmed network connectivity to the test environment prior to commencement of the conformance test. Participants must also ensure that a reasonable heartbeat interval greater than 10 seconds and no higher than 60 seconds in their Logon message. Recommended heartbeat interval is 30 seconds.	
#	Test Scenarios	Expected Results
1.	Establish a connection and logon using the Order Entry user with Message Type Logon (A).	Application receives an acknowledgment with Message Type Logon (A) confirming a successful log on, where SessionStatus (1409) = Session Active (0). Application will need to maintain heartbeats.

3.1.2. Logout

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can logout gracefully.	
Requirements	It is expected for the Participant to complete a graceful logout with the exchange.	
#	Test Scenarios	Expected Results
1.	Utilise a Logout (5) Message and disconnect from the session.	Application receives a Logout (5) message confirming a graceful log off.

3.1.3. Password Change

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can perform a password change as part of a logon to demonstrate to ASX that the application can manage its own password in day-to-day operations. The application must pass this test to progress to the next phase of functionality testing.	
Requirements	Participants must adhere to the ASX Password policy, outlined in ASX Trade FIX Order Entry specification under Password Complexity. ASX Trade FIX Order Entry Passwords expire every 90 days and must be changed by the Participant prior to this or it will not be able to logon to the Exchange.	
#	Test Scenarios	Expected Results
1.	Application should establish a connection and send a Logon (A) message where NewPassword (925) is the new application password that matches the ASX password policy.	Application should receive an acknowledgment with Message Type as Logon (A) confirming a successful Password change, where SessionStatus (1409) = Session Password Changed (1).

3.1.4. Sequence Reset

Category	Mandatory	
Description	The purpose of this test is to ensure applications can reset their own message sequence numbers. The application must pass this test to progress to the next phase of functionality testing.	
Requirements	Participants should send a Sequence Reset message with a value higher than their current sequence number.	
#	**Test Scenarios**	**Expected Results**
1.	Application performs an intra-day logon (A) message.	A logon acknowledgement (A) is sent by the exchange, confirming the session is active. Application will then need to maintain a heartbeat.
2.	After ensuring that ASX and the Participants application are exchanging heartbeats, applications should send a Sequence Reset message (4), specifically not in response to a Resend Request (2) from the exchange.	The next inbound message should have the MsgSeqNum (34) as the value specified in NewSeqNo (36) that was set in the prior Sequence Reset message. Application will need to maintain the Sequence numbers.

3.1.5. Reset Sequence Numbers

Category	Mandatory	
Description	The purpose of this test is to ensure the application can successfully complete an intra-day logon and reset its sequence numbers. The application must pass this test to progress to the next phase of functionality testing.	
Requirements	Use Logon (A) with ResetSeqNumFlag (141) set to Yes (Y) for intentional intra-day reset. Be aware: GTC/GTD long orders cannot be reloaded after a Logon message with ResetSeqNumFlag (141) set to Yes (Y) for an intra-day reset.	
#	**Test Scenarios**	**Expected Results**
1.	Complete a Logon (A) with ResetSeqNumFlag (141) = Yes (Y)	Application receives an acknowledgment with Message Type Logon(A) confirming a successful log on, where SessionStatus (1409) = Session Active (0). Application will then need to maintain a heartbeat.

3.2. TradeMatch Orders

3.2.1. Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a new limit order and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with Limit (2) order type.	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.2. Market-To-Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a new market-to-limit order and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new market-to-limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with ordtype (40) = Market-to-limit (K).	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.3. Short Sell Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new Short Sell order into a stock with Side (54) = Sell Short (5) and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Short Sell order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with Side (54) = Sell Short (5) and Ordtype (40) = Limit (2).	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.4. Best Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a new Best Limit Order and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Best Limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with Ordtype (40) = Pegged (P) and PegPriceType (1094) = Primary Peg (5).	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.5. Iceberg Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new Iceberg order into stock with a visible quantity and a total quantity and receive an order identifier that can be used to track the order. Participants can pick either order type and choose any stock.	
Requirements	Participants should send a new Iceberg order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with Ordtype (4) = Limit (2) or market-to-limit (K) and DisplayQty (1138) => 1000.	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.6. Auction Imbalance Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new Auction Imbalance order into any stock with the required details and receive an order identifier that can be used to track the order. Auction Imbalance orders are entered with a limit price and are only accepted with time validity Fill and Kill.	
Requirements	Participants should send a new Action Imbalance order and receive an execution report Viable session states for Auction Imbalance orders include: Pre_Open and Pre_CSPA.	
#	Test Scenarios	Expected Results
1.	The application will need to send a NewOrderSingle (D) message with Ordtype (4) = Limit with ExecInst (18) = Imbalance Only (i).	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.2.7. Order Modification

Category	Mandatory if any scenarios from 3.2 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully complete an order modification to any Trade Match order currently in the order book.	
Requirements	Participants should amend the order quantity or price.	
#	Test Scenarios	Expected Results
1.	The application will need to send an OrderCancelReplaceRequest (G) to modify the quantity or price of an existing order.	Receive ExecutionReport (8) with ExecType (150) = Replaced (5).

3.2.8. Order Cancellation

Category	Mandatory if any scenarios from 3.2 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully cancel a Trade Match order currently in the order book.	
Requirements	Participants should cancel an existing order.	
#	Test Scenarios	Expected Results
1.	The application will need to send an Order Cancel Request (F) to alter the quantity or price of an existing order.	Receive ExecutionReport (8) with OrdStatus (39) = Cancelled (4).

3.2.9. Cross Protocol/Cross Session Order Cancellation

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can support cross protocol/cross session cancellations.	
Requirements	The Participant should cancel an order entered via an alternative protocol and/or the Participant should cancel an order entered by a different Order Entry user under the same PID. Provide ASX with the security and OrderID of cancelled orders.	
#	Test Scenarios	Expected Results
1.	The application will need to send an OrderCancelRequest (F) to cancel an order currently on the order book sent from either an OUCH, OMNet or different fix user under the same PID.	Receive ExecutionReport (8) with OrdStatus (39) = Cancelled (4).

3.3. Centre Point Orders

For these scenarios Participants are encouraged to use securities with existing bid/offer spreads in TradeMatch to ensure there is a valid National Best Bid and Offer (NBBO).

3.3.1. Centre Point Market Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new Centre Point market order with the required details and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Centre Point market order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Market (1) and ExDestination (100) = ASXC.	Receive ExecutionReport (8) with OrdStatus (39) = New (0) and ExDestination (100) = ASXC.

3.3.2. Centre Point Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new Centre Point limit order with the required details and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Centre Point limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Pegged (P), PegPriceType (1094) = Mid-Price-Peg (2), and ExDestination (100) = ASXC	Receive ExecutionReport (8) with OrdStatus (39) = New (0) and ExDestination (100) = ASXC

3.3.3. Centre Point Dark Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a Centre Point dark limit order and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Centre Point Dark Limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Limit (2) and ExDestination (100) = ASXC	Receive ExecutionReport (8) with OrdStatus (39) = New (0), ExecType (150) = New (0) and ExDestination (100) = ASXC

3.3.4. Centre Point Order Entry with Minimum Acceptable Quantity

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter Centre Point Order Entry with MAQ and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Centre Point order with a MAQ and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with any OrdType (40), ExDestination (100) = ASXC and MinQty (110).	Receive ExecutionReport (8) with OrdStatus (39) = New (0), ExecType (150) = New (0) and ExDestination (100) = ASXC.

3.3.5. Centre Point Order Entry with Minimum Acceptable Quantity Single Fill

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a single fill Centre Point Order Entry with MAQ and receive an order identifier that can be used to track the order.	
Requirements	Participants should send a new Centre Point single fill order with a minimum acceptable quantity and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40), ExDestination (100) = ASXC, ExecInst (18) = Single Execution Requested (j) and MinQty (110).	Receive ExecutionReport (8) with OrdStatus (39) = New (0), ExecType (150) = New (0) and ExDestination (100) = ASXC.

3.3.6. Centre Point Order Modification

Category	Mandatory if any scenarios from 3.3 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully modify a Centre Point order currently in the order book.	
Requirements	Participants should amend the quantity or price of a Centre Point order.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a OrderCancelReplaceRequest (G) altering the quantity or price of an existing order with ExDestination (100) = ASXC.	Receive ExecutionReport (8) with OrdStatus (39) = New (0), ExecType (150) = Replaced (5) and ExDestination (100) = ASXC.

3.3.7. Centre Point Order Cancellation

Category	Mandatory if any scenarios from 3.3 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully cancel a Centre Point order currently in the order book.	
Requirements	Participants should cancel an existing Centre Point order.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a OrderCancelRequest (F) message resulting in the cancellation of a resting Centre Point order with ExDestination (100) = ASXC.	Receive ExecutionReport (8) with OrdStatus (39) = Cancelled (4), ExecType (150) = Cancelled (4) and ExDestination (100) = ASXC.

3.4. Sweep Order

For these scenarios Participants are encouraged to use securities with existing bid/offer spreads in TradeMatch to ensure there is a valid National Best Bid and Offer (NBBO).

3.4.1. Sweep Limit/Dual Post Order

Category	Supported Functionality
Description	The purpose of this test is to ensure that the application can successfully enter a sweep limit/dual post order and receive an order identifier that can be used to track the order.
Requirements	Participants should enter a sweep limit or dual post order and receive an execution report. A Sweep order may be entered as a Limit or a Market-to-Limit order. For a Market-to-Limit Sweep order, users will enter all order details except price.

	A Sweep Dual Posted Order is entered as a Sweep Order with the price set at Half Tick.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with ExDestination (100) = XASX.	Receive ExecutionReport (8) with ExecType (150) = New (0), OrdStatus (39) = New (0) and ExDestination (100) = XASX.

3.4.2. Sweep Limit Single Fill Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a Sweep Limit Single Fill order and receive an order identifier that can be used to track the order.	
Requirements	Participants should enter a Sweep Limit Single Fill order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Limit (2), ExecInst (18) = Single Execution Requested (j) and either Cancel on connection loss (o) or Reinstate on connection loss (n), ExDestination(100) = XASX and MinQty (110).	Receive ExecutionReport (8) with ExecType(150) = New (0), OrdStatus (39) = New (0) and ExDestination (100) = XASX.

3.4.3. Sweep Market-To-Limit Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a sweep Market-to-Limit order and receive an order identifier that can be used to track the order.	
Requirements	Participants should enter a Sweep Market-To-Limit order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Market-to-limit (K) and ExDestination (100) = XASX.	Receive ExecutionReport (8) with OrdType (40) = Market-to-limit (K), OrdStatus (39) = New (0) and ExDestination (100) = XASX.

3.4.4. Sweep Limit Iceberg order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a sweep Limit Iceberg order and receive an order identifier that can be used to track the order.	
Requirements	Participants should enter a Sweep Limit Iceberg order and receive an execution report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with OrdType (40) = Limit (2), DisplayQty (1138) and ExDestination (100) = XASX.	Receive ExecutionReport (8) with OrdType (40) = Limit (2), OrdStatus (39) = New (0), DisplayMethod (1084) = 2, ExecType (150) = New (2) and ExDestination (100) = XASX.

3.4.5. Sweep Order Modification

Category	Mandatory if any scenarios from 3.4 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully modify an existing Sweep order.	
Requirements	Participants should amend the quantity or price of a Sweep order.	
#	**Test Scenarios**	**Expected Results**

1.	The application will need to send an Order Cancel Replace Request (G) message altering the quantity or price of an existing order with ExDestination (100) = XASX.	Receive ExecutionReport (8) with OrdStatus (39) = New (0), ExecType (150) = Replaced (5) and ExDestination (100) = XASX.

3.4.6. Sweep Order Cancellation

Category	Mandatory if any scenarios from 3.4 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully cancel an existing Sweep order resting on the order book.	
Requirements	Participants should cancel an existing Sweep Order.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send an Order Cancel Request (F) message resulting in the cancellation of a resting sweep order with ExDestination (100) = XASX.	Receive ExecutionReport (8) with OrdStatus (39) = Cancelled (4), ExecType (150) = Cancelled (4) and ExDestination (100) = XASX.

3.5. Tailor Made Combination (TMC) Orders

3.5.1. TMC Creation

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully create a TMC utilising a Security Definition Request.	
Requirements	Participants should create a TMC.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a Security Definition Request (c) message with NoLegs (555) = 2 or more and SecurityRequestType (321) = RequestSecurityIdentityForSpecifications (1).	Receive a SecurityDefinition (d) message with SecurityRequestResult (560) = Valid Request (0).

3.5.2. TMC New Order

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully place a new order for a previously created TMC and receive an order identifier that can be used to track the order.	
Requirements	Participants should enter a new order utilising the TMC created in scenario 3.5.1 TMC Creation and receive an Execution Report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message with the TMC created in scenario 3.5.1 TMC Creation.	Receive ExecutionReport (8) with OrdStatus (39) = New (0) for previously created TMC.

3.5.3. TMC Order Modification

Category	Mandatory if any scenarios from 3.5 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully modify the previously placed TMC order.	
Requirements	Participants should amend the quantity or price for the TMC order created in scenario 3.5.2 TMC New Order.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send an Order Cancel Replace Request (G) message altering the quantity or price of the Order entered for 3.5.2 TMC New Order.	Receive ExecutionReport (8) with ExecType (150) = Replaced (5).

3.5.4. TMC Order Cancellation

Category	Mandatory if any scenarios from 3.5 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully cancel the previously placed order from 3.5.2 TMC New Order.	
Requirements	Participants should cancel the order created in scenario 3.5.2 TMC New Order.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send an Order Cancel Request (F) message resulting in the cancellation of the Order entered for 3.5.2 TMC New Order.	Receive ExecutionReport (8) with ExecType (150) = Cancelled (4).

3.6. Trade Reporting

3.6.1. Trade Capture Report Entry (Single-Sided)

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a Single Sided Trade Capture Report into the market.	
Requirements	Participants should enter a Single-Sided Trade Capture Report and receive a TradeCaptureReportAck.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a TradeCaptureReport (AE) message with TradeReportType (856) = Submit (0) and PartyRole (452) = Contra Firm (17).	Receive a TradeCaptureReportAck (AR) with TradeReportType (856) = Submit (0) and TrdRptStatus (939) = Accepted (0).

3.6.2. Trade Capture Report Entry (Two-Sided)

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a Two-Sided Trade Capture Report.	
Requirements	Participants should enter a Two-Sided Trade Capture Report and receive a TradeCaptureReportAck. For Two-Sided Trade Capture Reports SecurityType (167) = MLEG and NoLegs (555) needs to be 2 or more.	
#	**Test Scenarios**	**Expected Results**

1.	The application will need to send a TradeCaptureReport (AE) message with TradeReportType (856) = Submit (0), TradeHandlingInstr (1123) = Two party Report (1) and SecurityType (167) = MLEG.	Receive a TradeCaptureReportAck (AR) with TradeReportType (856) = Submit (0) and TrdRptStatus (939) = Accepted (0).

3.6.3. Trade Capture Report Entry (Initial Trade Notification)

Category	Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully enter a Two-Sided Trade Capture Report with an Initial Trade Notification (ITN) flag.	
Requirements	Participants should enter a Two-Sided Trade Capture Report with an ITN flag and receive a TradeCaptureReportAck. For Two Sided Trade Capture Reports, SecurityType (167) = MLEG and NoLegs (555) needs to be 2 or more.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a TradeCaptureReport (AE) message with TradeReportType (856) = Submit (0), TradeHandlingInstr (1123) = Two party Report(1), SecurityType (167) = MLEG and InitialTradeReportIndicator (24101) = Initial Trade Report (1).	Receive a TradeCaptureReportAck (AR) with TradeReportType (856) = Submit (0) and TrdRptStatus (939) = Accepted (0).

3.6.4. Trade Capture Report Cancellation

Category	Mandatory if any scenarios from 3.6 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully cancel an existing Single Sided Trade Capture Report.	
Requirements	Participants should cancel a Single Sided Trade Capture Report.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a TradeCaptureReport (AE) message with TradeReportType (856) = TradeReportCancel (6).	Receive a TradeCaptureReportAck (AR) with TradeReportType (856) = TradeReportCancel (6) and TrdRptStatus (939) = Cancelled (2).

3.7. Rejection Handling

3.7.1. Order Rejection Handling

Category	Mandatory if any scenarios from 3.2, 3.3, 3.4 or 3.5 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully handle order rejection messages.	
Requirements	Participants should enter orders matching the scenarios listed below and receive an Execution Report for each rejection outlined. Participants should pass all scenarios listed below.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a NewOrderSingle (D) message using an invalid Symbol (55) or SecurityID (48).	Receive ExecutionReport (8) with ExecType (150) = Rejected (8), OrdStatus (39) = Rejected (8) and description of rejection in RejectText (1328).

2.	The application will need to send a NewOrderSingle (D) message using an invalid OrdType (40).	Receive ExecutionReport (8) with ExecType (150) = Rejected (8), OrdStatus (39) = Rejected (8) and description of rejection in RejectText (1328).
3.	The application will need to send a NewOrderSingle (D) message using an invalid Price (44).	Receive Reject (3) message with RefTagID (371) = 44 (Price).

3.7.2. Trade Capture Report Rejection

Category	Mandatory if any scenarios from 3.6 are Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully handle Trade Report rejection messages.	
Requirements	Participants should enter Trade Capture Reports matching the scenarios listed below and receive a TradeCaptureReportAck for each rejection outlined. Participants should pass all scenarios listed below.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a TradeCaptureReport (AE) with invalid security details for Symbol (55).	Receive TradeCaptureReportAck (AR) with TrdRptStatus (939) = Rejected (1) and TradeReportRejectReason (751) = Unknown Instrument (2).
2.	The application will need to send a TradeCaptureReport (AE) with Invalid Trade Report Code (24105).	Receive TradeCaptureReportAck (AR) with TrdRptStatus (939) = Rejected (1) and TradeReportRejectReason (751) = Other ASX extension (99).
3.	The application will need to send a TradeCaptureReport (AE) with Invalid Trade Report Type (856).	Receive TradeCaptureReportAck (AR) with TrdRptStatus (939) = Rejected (1) and TradeReportRejectReason (751) = Invalid Trade Type (4).

3.7.3. Business Message Rejection

Category	Mandatory if scenario 3.8.4 is Supported Functionality	
Description	The purpose of this test is to ensure that the application can successfully handle a BusinessMessageReject (j) message.	
Requirements	Participants should enter a OrderStatusRequest (H) message that triggers a Business Message Reject (j) response.	
#	**Test Scenarios**	**Expected Results**
1.	The application will need to send a OrderStatusRequest (H) message with Invalid CliOrderID (no matching order in orderbook).	Receive a BusinessMessageReject (j) with relevant BusinessRejectReason (380).

3.8. Additional Attributes

3.8.1. Cancel On Disconnect

Category	Supported Functionality
Description	The purpose of this test is to ensure that the application can successfully submit a new order with the Cancel on Disconnect functionality enabled.
Requirements	Participants should send a new order with ExecInst (18) set to Cancel On Connection Loss (o) and receive an execution report.

#	Test Scenarios	Expected Results
1.	The application will need to send a NewOrderSingle (D) message with ExecInst (18) = Cancel on connection loss (o).	Receive ExecutionReport (8) with OrdStatus (39) = New (0).

3.8.2. Unintentional Crossing Prevention (UCP)

Category	Supported Functionality
Description	The purpose of this test is to ensure that the application can successfully trigger an execution report indicating that an order has matched internally via utilising a CrossingKey (24111).
Requirements	Participants should place buy and sell orders with a CrossingKey so they match each other and generate Execution Reports.

#	Test Scenarios	Expected Results
1.	The application will need to send a buy side NewOrderSingle (D) message with CrossingKey (24111) populated.	Receive ExecutionReport (8) with tag OrdStatus (39) = New (0).

3.8.3. Mass Cancel Request

Category	Supported Functionality
Description	The purpose of this test is to ensure that the application can successfully cancel a group of previously placed orders with a OrderMassCancelRequest (q) message.
Requirements	Participants should enter an Order Mass Cancel Request for an active order and receive an Order Mass Cancel Report.

#	Test Scenarios	Expected Results
1.	The application will need to send a OrderMassCancelRequest (q) message with MassCancelRequestType (530) value of Cancel orders for a security (1) or Cancel orders for an Underlying security (Instrument underlying) (2).	Receive a OrderMassCancelReport (r) message MassCancelResponse (531) value of Cancel orders for security (1), Cancel orders for an Underlying security (2) or Cancel orders for an Instrument Class (5).

3.8.4. Order Status Request

Category	Supported Functionality
Description	The purpose of this test is to ensure that the application can send an Order Status Request (H) message to query for the current state of an order.
Requirements	Participants should send an Order Status Request and receive the relevant execution reports for the requested order. Users can utilise either the OrderId (37) or ClOrdId (11) value to request the status of the order.

#	Test Scenarios	Expected Results
1.	The Participant enters GTC/GTD orders for any security.	Users will receive an Execution Report (8) with OrdStatus (39) = 0 (New).
2.	The Participant to send an Order Status Request (H).	The application will receive Execution Report/s (8) correlating to the original order.

4. ASX Assisted Conformance Scenarios

This phase of the conformance test process is allocated for one-to-one testing between the Participant and a member of ASX Customer Technical Support. These scenarios will be tested in a controlled manner with both sides on the phone confirming each test in turn independently.
This session should only be booked with ASX when the "Self-testing" phase is successfully completed.

4.1. Logon (Application Name and Version Validation)

Category	Mandatory	
Description	The purpose of this test is to verify whether the Application name and version (Tag 1408) in Logon (A) message matches the details provided on the conformance checklist.	
Requirements	Participants should perform a Logon (A) with ResetSeqNumFlag (141) = No (N).	
#	**Test Scenarios**	**Expected Results**
1.	Participant should send Logon (A) message with DefaultCstmApplVerID (1408) correctly populated.	ASX to verify if the details supplied in the logon messages matches with the value supplied in the conformance checklist.

4.2. Account Disable

Category	Mandatory	
Description	The purpose of this test is to validate the behaviour of the Participants application once it has been logged out forcefully and disabled to ensure it does not attempt multiple logins.	
Requirements	Participants should logon and maintain their session. Post being logged out by the ASX, Participants applications should attempt to logon no more than 10 times with a 5 second delay between attempts. Once instructed to do so the Participant should complete a Logon (A) and maintain the session.	
#	**Test Scenarios**	**Expected Results**
1.	ASX will force logout and disable the session, subsequently triggering a Logout (5) message.	The application will receive a Logout (5) message with SessionStatus (1409) = Session Logout Complete (4).
2.	The application will perform another Logon (A) while the sessionremains disabled.	The application will receive a Logout (5) message with SessionStatus (1409) = Account Locked (6). The application is observed and should not be attempting multiple Logon (A) attempts once the session is locked.
3.	ASX will re-enable the session and advise the Participant to initiate another Logon (A).	The application receives Logon (A) Message confirming a successful logon, where SessionStatus (1409) = Session Active (0). The application will then need to maintain a heartbeat.

4.3. Password Expiry

Category	Mandatory
Description	The purpose of this test is to ensure an application can correctly manage a session that has reached its password expiry date. The Participant will receive an error message upon logon, relating to a session that requires a change in password. ASX will force the session into an expired password state and the application should re-establish connectivity and provide a new password within the Logon (A) message. It is expected that the application will refrain from any further connection attempts if it's unable to provide a valid logon message after the password expires.

Requirements	Participants should logon and resolve an expired password prompt by completing a Logon (A) message with a valid NewPassword (925) tag and receive a Logon response (A) from the ASX with SessionStatus (1409) = Session password changed (1) and then maintain the session.	
#	**Test Scenarios**	**Expected Results**
1.	The application will perform a standard Logout (5).	The application will receive a Logout (5) confirmation.
2.	ASX will set a new password which will be configured to expire immediately, then advise the Participant of the new password. The application should send a Logon (A) message using the new password provided by ASX in Password (554) field.	The application will receive a Logout (5) from the exchange where the SessionStatus (1409) = Password expired (8) and its connection will be terminated by the exchange.
3.	The application will then need to send a Logon (A) message, where it sends a valid NewPassword (925) value.	The application will receive a Logon (A) message with SessionStatus (1409) = Session password changed (1) and ensure it maintains a heartbeat.

4.4. Unsolicited Order Cancellation

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can receive an Execution Report (8) with OrdStatus (39) = Cancelled (4) when ASX performs an unsolicited order cancellation.	
Requirements	Participants should enter 3 seperate orders, The ASX will cancel one of these orders and the Participant will need to provide requested details for the cancelled order.	
#	**Test Scenarios**	**Expected Results**
1.	The Participant enters 3 separate orders for the same security of the Participants choosing.	Users will receive an Execution Report (8) with OrdStatus (39) = New (0).
2.	ASX will cancel the order and Participant to confirm details of the order.	Users will receive an Execution Report (8) with OrdStatus (39) = 4 (Cancelled).

4.5. Good-Till-Cancelled (GTC)/Good-To-Date (GTD) Order Recovery

Category	Mandatory if utilising GTC/GTD orders	
Description	The purpose of this test is to ensure that the application can receive Execution Reports (8) of GTC/GTD orders sent on a trading date prior to the commencement of the Assisted Test.	
Requirements	Participant should send GTC/GTD orders with an security on a trading date prior to the assisted test which remain active. Participants should Logon (A) during the assisted test and provide requested details for the chosen recovered order.	
#	**Test Scenarios**	**Expected Results**
1.	The application to perform a Logon (A) with ResetSeqNumFlag (141) = No (N).	Users will receive an acknowledgement from ASX for Logon (A) followed by the receipt of GTC/GTD Orders with Execution Reports (8). User must maintain session during order recovery.

4.6. Unsolicited Trade Cancellation

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can receive an unsolicited Trade Cancellation message. Unsolicited Trade cancellations initiated by the ASX will result in the application receiving a Trade Capture Report (AE) containing details of the Trade Cancel.	
Requirements	Participants should enter an order with an security of their choosing. ASX will execute against this order to generate a trade which will subsequently be cancelled. Participants should then provide requested details for the cancelled trade.	
#	**Test Scenarios**	**Expected Results**
1.	The Participant enters an order for any security.	Users will receive an Execution Report (8) with OrdStatus (39) = 0 (New).
2.	ASX will execute against the order to generate a trade that will either partially or fully fill the order.	Users will receive an Execution Report (8) with OrdStatus (39) = 1 (partially filled) or 2 (fully filled).
3.	ASX will cancel the resulting trade.	Users will receive a Trade Capture Report (AE) with ExecType (150) = H (Trade Cancel) and provide requested details to the ASX.

4.7. Resend Request

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can send a Resend Request (2) to initiate the retransmission of messages intra-day.	
Requirements	Participants should enter new orders with a security of their choosing and then complete a graceful Logout (5). ASX will then cancel the order/s. Participants should then send a Logon (A) message ensuring ResetSeqNumFlag (141) = No (N). Participants should then initiate a Resend Request (2) and receive relevant Execution Reports.	
#	**Test Scenarios**	**Expected Results**
1.	The Participant enters orders for any security.	Users will receive an Execution Report (8) with OrdStatus (39) = 0 (New).
2.	The Participant to gracefully Logout (5), ASX will then cancel the orders.	The application will receive a Logout (5) confirmation.
3.	Participant to send Logon (A) message with ResetSeqNumFlag (141) set to ‘N’, followed by a Resend Request (2) from the session for the missing sequence numbers.	The application will receive an Execution Report/s (8) of cancelled orders and is required to maintain the session.

4.8. Failover

Category	Mandatory	
Description	The purpose of this test is to ensure that the application can recover from a failover event and re-establish connectivity to the session using the next expected sequence number. Please refer to the ASX Trade Connectivity Guide for further information regarding port details for ASX Trade FIX Order Entry.	
Requirements	The Participant should complete a graceful Logout (5) and change the port they are targeting. The Participant should then complete a Logon (A) to the new port and maintain heartbeats.	
#	**Test Scenarios**	**Expected Results**

1.	The Participant to gracefully Logout (5) from session.	The application will receive a Logout (5) confirmation.
2.	The Participant to connect to an alternate port with ResetSeqNumFlag (141) set to ‘N’ whilst maintaining sequence number from the previous session.	ASX will respond with Logon (A) message.