
**Financial services — Financial
information eXchange session layer —
Part 3:
FIX session layer test cases**

STANDARDSISO.COM : Click to view the full PDF of ISO 3531-3:2022



STANDARDSISO.COM : Click to view the full PDF of ISO 3531-3:2022



COPYRIGHT PROTECTED DOCUMENT

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 FIX session layer test cases	1
4.1 Applicability	1
4.2 Test cases	1
4.3 Buyside-oriented (session initiator) Logon and session initiation test case	1
4.3.1 Scenario 1B Connect and Send Logon message	1
4.4 Sellside-oriented (session acceptor) Logon and session initiation test case	2
4.4.1 Scenario 1S Receive Logon message	2
4.4.2 Scenario 2S. Receive any message other than a Logon message	3
4.5 Test cases applicable to all FIX systems	3
4.5.1 Scenario 2 Receive Message Standard Header	3
4.5.2 Scenario 3 Receive Message Standard Trailer	5
4.5.3 Scenario 4 Send Heartbeat message	6
4.5.4 Scenario 5 Receive Heartbeat message	6
4.5.5 Scenario 6 Send Test Request	6
4.5.6 Scenario 7 Receive Reject message	6
4.5.7 Scenario 8 Receive Resend Request message	7
4.5.8 Scenario 9 Synchronize sequence numbers	7
4.5.9 Scenario 10 Receive Sequence Reset (Gap Fill)	7
4.5.10 Scenario 11 Receive Sequence Reset (Reset)	8
4.5.11 Scenario 12 Initiate logout process	8
4.5.12 Scenario 13 Receive Logout message	9
4.5.13 Scenario 14 Receive application or session layer message	9
4.5.14 Scenario 15 Send application or session layer messages to test normal and abnormal behaviour/response	11
4.5.15 Scenario 16 Queue outgoing messages	11
4.5.16 Scenario 17 Support encryption	12
4.5.17 Scenario 18 Support third party addressing	13
4.5.18 Scenario 19 Test PossResend handling	14
4.5.19 Scenario 20 Simultaneous Resend request test	14
Bibliography	15

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by FIX Trading Community (as FIX Session Layer Technical Specification) and drafted in accordance with its editorial rules. It was assigned to Technical Committee ISO/TC 68, *Financial services*, Subcommittee SC 9, *Information exchange for financial services*, and adopted under the “fast-track procedure”.

A list of all parts in the ISO 3531 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

FIX session protocol was written to be independent of any specific communications protocol (e.g. X.25, async, TCP/IP) or physical medium (e.g. copper, fibre, satellite) chosen for electronic data delivery. It offers a reliable stream where a message is delivered once and in order. The FIX session layer is designed to survive and resume operation in the event of the loss of transport level connections caused by any type of failure, including network outage, application failure or computer hardware failures.

The session layer is concerned with the ordered delivery of data while the application level defines business-related data content. This document focuses on the ordered delivery of data using the “FIX session protocol”.

The FIX session protocol is implemented using the FIX tagvalue encoding syntax for the standard header, standard trailer and the session level messages which make up the FIX session protocol. It is possible to send messages using other FIX-defined encodings (e.g. FIXML, SBE, JSON, GPB, ASN.1) or other non-FIX defined encodings (e.g. XML, FpML, ISO 20022 XML, JSON).

STANDARDSISO.COM : Click to view the full PDF of ISO 3531-3:2022

STANDARDSISO.COM : Click to view the full PDF of ISO 3531-3:2022

Financial services — Financial information eXchange session layer —

Part 3: FIX session layer test cases

1 Scope

This document provides a set of mandatory and optional conformity tests applicable to all versions of the FIX session layer standard.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11404, *Information technology — General-Purpose Datatypes (GPD)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11404 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

4 FIX session layer test cases

4.1 Applicability

This document was last revised September 20, 2002 at which time FIX version [4.3](#) with Errata 20020930 was the latest version of the FIX Protocol. Note that future amendments to this document can be found on the FIX website and any version of this document published on a later date takes precedence over this version of the document. This document is applicable to all supported versions of the FIX session layer ([4.2](#), [4.4](#), FIXT) except where explicitly indicated.

4.2 Test cases

These test cases are from the perspective of the FIX system being tested. The FIX system receives the “condition/stimulus” and is expected to take the appropriate action as defined by “expected behaviour”.

4.3 Buyside-oriented (session initiator) Logon and session initiation test case

4.3.1 Scenario 1B Connect and Send Logon message

Mandatory

Condition/stimulus	Expected behaviour
a. Establish transport layer connection.	Successfully established transport layer connection with peer.
b. Send Logon(35=A) request message.	Logon(35=A) acknowledgement message sent by peer.
c. Valid Logon(35=A) acknowledgement message received.	If MsgSeqNum(34) is too high, send ResendRequest(35=2).
d. Invalid Logon(35=A) acknowledgement message received.	1. Generate an error condition in test output. 2. (Optional) Send Reject(35=3) message with RefSeqNum(45) identifying Logon(35=A) message's MsgSeqNum(34) and Text(58) referencing error condition. 3. Send Logout(35=5) message with Text(58) referencing error condition. 4. Disconnect.
e. Receive any message other than a Logon(35=A) message.	1. Log an error "first message not a logon". 2. (Optional) Send Reject(35=3) message with RefSeqNum(45) identifying message's MsgSeqNum(34) and Text(58) referencing error condition. 3. (Optional) Send Logout(35=5) message with Text(58) referenceing error condition. 4. Disconnect.

4.4 Sellside-oriented (session acceptor) Logon and session initiation test case

4.4.1 Scenario 1S Receive Logon message

Mandatory

Condition/stimulus	Expected behaviour
a. Valid Logon(35=A) request message received.	1. Respond with Logon(35=A) acknowledgement message. 2. If MsgSeqNum(34) > NextNumIn send ResendRequest(35=2).
b. Logon(35=A) message received with duplicate identity (e.g. same IP, port, SenderCompID(49), TargetCompID(56), etc. as existing connection).	1. Generate an error condition in test output. 2. Disconnect without sending a message (Note: sending a Reject or Logout(35=5) would consume a MsgSeqNum(34)).
c. Logon(35=A) message received with unauthenticated/non-configured identity (e.g. invalid SenderCompID(49), invalid TargetCompID(56), invalid source IP address, etc. vs. system configuration).	1. Generate an error condition in test output. 2. Disconnect without sending a message (Note: sending a Reject or Logout(35=5) would consume a MsgSeqNum(34)).

Condition/stimulus	Expected behaviour
d. Invalid Logon(35=A) message.	1. Generate an error condition > in test output. 2. (Optional) Send > Reject(35=3) message > with RefSeqNum(45) > identifying Logon(35=A) > message's MsgSeqNum(34) > with Text(58) referencing > error condition. 3. Send Logout(35=5) message > with Text(58) referencing > error condition. 4. Disconnect.

4.4.2 Scenario 2S. Receive any message other than a Logon message

Mandatory

Condition/stimulus	Expected behaviour
First message received is not a Logon(35=A) message.	1. Log an error "First message not a logon". 2. Disconnect.

4.5 Test cases applicable to all FIX systems

4.5.1 Scenario 2 Receive Message Standard Header

Mandatory

Condition/stimulus	Expected behaviour
a. MsgSeqNum(34) received as expected.	Accept MsgSeqNum(34) for the message.
b. MsgSeqNum(34) higher than expected.	Respond with ResendRequest(35=2) message.
c. MsgSeqNum(34) lower than expected without PossDupFlag(43) set to Y. Exception: SequenceReset(35=4).	1. Whenever possible it is recommended that FIX engine attempt to send a Logout(35=5) message with a text message of "MsgSeqNum too low, expecting X but received Y". 2. (Optional) Wait for Logout(35=5) message response (Note: likely will have inaccurate MsgSeqNum(34)) or wait 2 seconds, whichever comes first. 3. Disconnect. 4. Generate an error condition in test output.
d. Garbled message received.	1. Consider garbled and ignore message (do not increment NextNumIn) and continue accepting messages. 2. Generate a warning condition in test output.
e. PossDupFlag(43) set to Y; OrigSendingTime(122) specified is less than or equal to SendingTime(52) and MsgSeqNum(34) lower than expected. Note: OrigSendingTime(122) should be earlier than SendingTime(52) unless the message is being resent within the same second during which it was sent.	1. Check to see if MsgSeqNum(34) has already been received. 2. If already received then ignore the message, otherwise accept and process the message.

Condition/stimulus	Expected behaviour
f. PossDupFlag(43) set to Y; OrigSendingTime(122) specified is greater than SendingTime(52) and MsgSeqNum(34) as expected. Note: OrigSendingTime(122) should be earlier than SendingTime(52) unless the message is being resent within the same second during which it was sent.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 10 (SendingTime accuracy problem). 2. Increment NextNumIn. 3. Optional: • Send Logout(35=5) message referencing inaccurate SendingTime(52) value. • (Optional) Wait for Logout(35=5) message response (Note: likely will have inaccurate SendingTime(52)) or wait 2 seconds, whichever comes first. • Disconnect. Generate an error condition in test output.
g. PossDupFlag(43) set to Y and OrigSendingTime(122) not specified. Note: Always set OrigSendingTime(122) to the time when the message was originally sent-not the present SendingTime(52) and set PossDupFlag(43)=Y when responding to a ResendRequest(35=2).	1. Send Reject(35=3) message with SessionRejectReason(373) set to 1 (Required tag missing). 2. Increment NextNumIn.
h. BeginString(8) value received as expected and specified in testing profile and matches BeginString(8) on outbound messages.	Accept BeginString(8) for the message.
i. BeginString(8) value received did not match value expected and specified in testing profile or does not match BeginString(8) on outbound messages.	1. Send Logout(35=5) message referencing incorrect BeginString(8) value. 2. (Optional) Wait for Logout(35=5) message response (Note: likely will have incorrect BeginString(8)) or wait LogoutAckThreshold seconds, whichever comes first. 3. Disconnect. 4. Generate an error condition in test output.
j. SenderCompID(49) and TargetCompID(56) values received as expected and specified in testing profile.	Accept SenderCompID(49) and TargetCompID(56) for the message.
k. SenderCompID(49) and TargetCompID(56) values received did not match values expected and specified in testing profile.	1. Send Reject(35=3) message with RefTagID(371) to identify field with mismatched value, and SessionRejectReason(373) set to 9 (CompID problem). 2. Increment NextNumIn. 3. Send Logout(35=5) message referencing incorrect SenderCompID(49) or TargetCompID(56) value. 4. (Optional) Wait for Logout(35=5) message response (Note: likely will have incorrect SenderCompID(49) or TargetCompID(56)) or wait 2 seconds, whichever comes first. 5. Disconnect. 6. Generate an error condition in test output.

Condition/stimulus	Expected behaviour
l. BodyLength(9) value received is correct.	Accept BodyLength(9) for the message.
m. BodyLength(9) value received is not correct.	1. Consider garbled and ignore message (do not increment NextNumIn) and continue accepting messages. 2. Generate a warning condition in test output.
n. SendingTime(52) value received is specified in UTC (Universal Time Coordinated) also known as GMT) or is not within <i>SendingTimeThreshold</i> seconds of a synchronized time source.	Accept SendingTime(52) for the message.
o. SendingTime(52) value received is either not specified in UTC (Universal Time Coordinated) or is not GMT) or is not within a within <i>SendingTimeThreshold</i> seconds of a synchronized time source. Rationale: Verify system clocks on both sides are in sync and that SendingTime(52) is current time.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 10 (SendingTime accuracy problem). 2. Increment NextNumIn. 3. Send Logout(35=5) message referencing inaccurate SendingTime(52) value. 4. (Optional) Wait for Logout(35=5) message response (Note: likely will have inaccurate SendingTime(52)) or wait 2 seconds, whichever comes first. 5. Disconnect. 6. Generate an error condition in test output.
p. MsgType(35) value received is valid (defined in spec or classified as user-defined).	Accept MsgType(35) for the message.
q. MsgType(35) value received is not valid (defined in spec or classified as user-defined).	1. Send Reject(35=3) message with SessionRejectReason(373) set to 11 (Invalid MsgType). 2. Increment NextNumIn. 3. Generate a warning condition > in test output.
r. MsgType(35) value received is valid (defined in spec or classified as user-defined) but not supported or registered in testing profile.	1. Send BusinessMessageReject(35=j) with BusinessRejectReason(380) set to 3 (Unsupported Message Type). 2. Increment NextNumIn. 3. Generate a warning condition in test output.
s. BeginString(8), BodyLength(9), and MsgType(35) are first three fields of message.	Accept the message.
t. BeginString(8), BodyLength(9), and MsgType(35) are not the first three fields of message.	1. Consider garbled and ignore message. 2. Do not increment NextNumIn. 3. Continue accepting messages. 4. Generate a warning condition in test output.

4.5.2 Scenario 3 Receive Message Standard Trailer

Mandatory

Condition/stimulus	Expected behaviour
a. Valid CheckSum(10).	Accept message

Condition/stimulus	Expected behaviour
b. Invalid CheckSum(10).	1. Consider garbled and ignore message. 2. Do not increment NextNumIn. 3. Continue accepting messages. 4. Generate a warning condition in test output.
c. Garbled message.	1. Consider garbled and ignore message. 2. Do not increment NextNumIn. 3. Continue accepting messages. 4. Generate a warning condition in test output.
d. CheckSum(10) is last field of message, value has length of 3, and is delimited by <SOH>.	Accept message.
e. CheckSum(10) is not the last field of message, value does not have length of 3, or is not delimited by <SOH>.	1. Consider garbled and ignore message. 2. Do not increment NextNumIn. 3. Continue accepting messages. 4. Generate a warning condition in test output.

4.5.3 Scenario 4 Send Heartbeat message

Mandatory

Condition/stimulus	Expected behaviour
a. No data sent during preset heartbeat interval (HeartBtInt(108) field).	Send Heartbeat(35=0) message.
b. TestRequest(35=1) message received.	Send Heartbeat(35=0) message with TestRequest(35=1) message's TestReqID(112).

4.5.4 Scenario 5 Receive Heartbeat message

Mandatory

Condition/stimulus	Expected behaviour
Valid Heartbeat(35=0) message.	Accept Heartbeat(35=0) message.

4.5.5 Scenario 6 Send Test Request

Mandatory

Condition/stimulus	Expected behaviour
No data received during preset heartbeat interval (HeartBtInt(108)) + "some reasonable period of time" (use 20% of HeartBtInt(108)).	1. Send TestRequest(35=1) message. 2. Track and verify that a Heartbeat(35=0) message with the same TestReqID(112) is received (may not be the next message received).

4.5.6 Scenario 7 Receive Reject message

Mandatory

Condition/stimulus	Expected behaviour
Valid Reject(35=3) message.	1. Increment NextNumIn. 2. Continue accepting messages.

4.5.7 Scenario 8 Receive Resend Request message

Mandatory

Condition/stimulus	Expected behaviour
Valid ResendRequest(35=2).	Respond with application layer messages and SequenceReset(35=4) with GapFillFlag(123)=Y for session layer messages in request range according to message recovery rules.

4.5.8 Scenario 9 Synchronize sequence numbers

Optional

Condition/stimulus	Expected behaviour
Application failure that caused loss of session state.	1. Establish new FIX connection. 2. Reset NextNumOut to an arbitrarily large value that is greater than the last known NextNumOut value. 3. Reset NextNumIn value to 1. 4. Send SequenceReset(35=4) with GapFillFlag(123)=N and NewSeqNo(36) set to a number arbitrarily larger than the last known NextNumOut value. 5. Process retransmitted messages and gap fills from peer. OR 1. Both peers reset their NextNumIn and NextNumOut values to 1. 2. Start new FIX session

4.5.9 Scenario 10 Receive Sequence Reset (Gap Fill)

Mandatory

Condition/stimulus	Expected behaviour
a. Receive SequenceReset(35=4) with GapFillFlag(123)=Y message with NewSeqNo(36) > MsgSeqNum(34) and MsgSeqNum(34) > NextNumIn	Issue ResendRequest(35=2) to fill gap between last expected MsgSeqNum(34) & received MsgSeqNum(34).
b. Receive SequenceReset(35=4) with GapFillFlag(123)=Y message with NewSeqNo(36) > MsgSeqNum(34) and MsgSeqNum(34) = NextNumIn	Set next expected sequence number = NewSeqNo(36).

Condition/stimulus	Expected behaviour
c. Receive SequenceReset(35=4) with GapFill-Flag(123)=Y message with NewSeqNo(36) > MsgSeqNum(34) and MsgSeqNum(34) < NextNumIn and PossDupFlag(43)=Y.	Ignore message.
d. Receive SequenceReset(35=4) with GapFill-Flag(123)=Y message with NewSeqNo(36) > MsgSeqNum(34) and MsgSeqNum(34) < NextNumIn and without PossDupFlag(43)=Y.	1. If possible, send a Logout(35=5) message with text of "MsgSeqNum too low, expecting X received Y," prior to disconnecting FIX session. 2. (Optional) Wait for Logout(35=5) message response (Note: likely will have inaccurate MsgSeqNum(34)) or wait 2 seconds, whichever comes first. 3. Disconnect. 4. Generate an error condition in test output.
e. Receive SequenceReset(35=4) with GapFill-Flag(123)=Y message with NewSeqNo(36) <= MsgSeqNum(34) and MsgSeqNum(34) = NextNumIn	Send Reject(35=3) message with message "attempt to lower sequence number, invalid value NewSeqNo(36)=<x>".

4.5.10 Scenario 11 Receive Sequence Reset (Reset)

Mandatory

Condition/stimulus	Expected behaviour
a. Receive SequenceReset(35=4) with GapFill-Flag(123)=N message with NewSeqNo(36) > NextNumIn	1. Accept the SequenceReset(35=4) message without regard to its MsgSeqNum(34). 2. Set NextNumIn to NewSeqNo(36).
b. Receive SequenceReset(35=4) with GapFill-Flag(123)=N message with NewSeqNo(36) = NextNumIn	1. Accept the SequenceReset(35=4) message without regard to its MsgSeqNum(34). 2. Generate a warning condition in test output.
c. Receive SequenceReset(35=4) with GapFill-Flag(123)=N message with NewSeqNo(36) < NextNumIn	1. Accept the SequenceReset(35=4) message without regard to its MsgSeqNum(34). 2. Send Reject(35=3) message with SessionRejectReason(373) set to 5 (Value is incorrect (out of range) for this tag). 3. Do NOT change the value of NextNumIn. 4. Generate an error condition in test output.

4.5.11 Scenario 12 Initiate logout process

Mandatory

Condition/stimulus	Expected behaviour
Initiate logout.	1. Send Logout(35=5) message. 2. Wait for counterparty to respond with Logout(35=5) message <i>LogoutAckThreshold</i> seconds (Note: may not be received if communications problem exists). If not received, generate a warning condition in test output. 3. Disconnect.

4.5.12 Scenario 13 Receive Logout message

Mandatory

Condition/stimulus	Expected behaviour
a. Receive valid Logout(35=5) acknowledgement in response to a Logout(35=5) request.	Disconnect without sending a message.
b. Receive valid Logout(35=5) request message unsolicited.	1. Send Logout(35=5) acknowledgement message. 2. Wait for counterparty to disconnect up to 10 seconds. If max exceeded, disconnect and generate an error condition in test output.

4.5.13 Scenario 14 Receive application or session layer message

Mandatory

Condition/stimulus	Expected behaviour
a. Receive field identifier (tag number) not defined in specification. Exception: undefined tag used is specified in testing profile as user-defined.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 0 (Invalid tag number). 2. Increment NextNumIn. 3. Generate an error condition in test output.
b. Receive message with a required field identifier (tag number) missing.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 1 (Required tag missing). 2. Increment NextNumIn. 3. Generate an error condition in test output.
c. Receive message with field identifier (tag number) which is defined in the specification but not defined for this message type. Exception: undefined tag used is specified in testing profile as user-defined for this message type.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 2 (Tag not defined for this message type). 2. Increment NextNumIn. 3. Generate an error condition in test output.
d. Receive message with field identifier (tag number) specified but no value (e.g. "55=<SOH>" vs. "55=IBM<SOH>").	1. Send Reject(35=3) message with SessionRejectReason(373) set to 4 (Tag specified without a value). 2. Increment NextNumIn. 3. Generate an error condition in test output.

Condition/stimulus	Expected behaviour
e. Receive message with incorrect value (out of range or not part of valid list of enumerated values) for a particular field identifier (tag number). Exception: undefined enumeration values used are specified in testing profile as user-defined.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 5 (Value is incorrect (out of range) for this tag). 2. Increment inbound MsgSeqNum(34). 3. Generate an error condition in test output.
f. Receive message with a value in an incorrect data format (syntax) for a particular field identifier (tag number).	1. Send Reject(35=3) message with SessionRejectReason(373) set to 6 (Incorrect data format for value). 2. Increment NextNumIn. 3. Generate an error condition in test output.
g. Receive a message in which the following is not true: Standard Header fields appear before Body fields which appear before Standard Trailer fields.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 14 (Tag specified out of required order). 2. Increment NextNumIn. 3. Generate an error condition in test output.
h. Receive a message in which a field identifier (tag number) which is not part of a repeating group is specified more than once.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 13 (Tag appears more than once). 2. Increment NextNumIn. 3. Generate an error condition in test output.
i. Receive a message with repeating groups in which the "count" field value for a repeating group is incorrect.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 16 (Incorrect NumInGroup count for repeating group). 2. Increment NextNumIn. 3. Generate an error condition in test output.
j. Receive a message with repeating groups in which the order of repeating group fields does not match the specification.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 15 (Repeating group fields out of order). 2. Increment NextNumIn 3. Generate an error condition in test output.
k. Receive a message with a field of a data type other than "data" which contains one or more embedded <SOH> values.	1. Send Reject(35=3) message with SessionRejectReason(373) (Non "data" value includes field delimiter (<SOH> character)). 2. Increment NextNumIn 3. Generate an error condition in test output. OR 1. Consider garbled and ignore message. 2. Generate an error condition in test output
l. Receive a message when application layer processing or system is not available (optional).	1. Send BusinessMessageReject(35=j) with BusinessRejectReason(380) set to 4 (Application not available). 2. Increment NextNumIn. 3. Generate a warning condition in test output.

Condition/stimulus	Expected behaviour
m. Receive a message in which a conditionally required field is missing.	1. Send BusinessMessageReject(35=j) with BusinessRejectReason(380) set to 5 (Conditionally Required Field Missing). 2. Increment NextNumIn. 3. Generate an error condition in test output.
n. Receive a message in which a field identifier (tag number) appears in both cleartext and encrypted section but has different values.	1. Send Reject(35=3) message with SessionRejectReason(373) set to 7 (Decryption problem). 2. Increment NextNumIn. 3. Generate an error condition in test output.

4.5.14 Scenario 15 Send application or session layer messages to test normal and abnormal behaviour/response

Optional

Condition/stimulus	Expected behaviour
Send more than one message of the same type with header and body fields ordered differently to verify acceptance. (Exclude those which have restrictions regarding order.)	Message accepted and subsequent messages' MsgSeqNum(34) are accepted.

4.5.15 Scenario 16 Queue outgoing messages

Mandatory

Condition/stimulus	Expected behaviour
a. Message to send/queue while disconnected.	Queue outgoing messages. Note there are two valid approaches: 1. Queue without regard to MsgSeqNum(34). a. Store data for messages 2. Queue each message with the next MsgSeqNum(34) value. b. Store data for messages in such a manner as to use and "consume" the next MsgSeqNum(34). Note: SendingTime(52) must contain the time the message is sent, not the time the message was queued.