BS ISO 3531-2:2022

BSI Standards Publication

Financial services — Financial information eXchange session layer

Part 2: FIX session layer

Documents
Document Sharing Hub

Full document details: BS ISO 3531-2:2022

National foreword

This British Standard is the UK implementation of ISO 3531-2:2022.

The UK participation in its preparation was entrusted to Technical Committee IST/12, Financial services.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 14994 4

ICS 03.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2022.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 3531-2:2022

INTERNATIONAL STANDARD

ISO 3531-2

First edition
2022-04

Financial services — Financial information eXchange session layer —

Part 2: FIX session layer

Reference number
ISO 3531-2:2022(E)

Documents
Document Sharing Hub
© ISO 2022

Full document details: BS ISO 3531-2:2022

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

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 3531-2:2022

Table of Contents

Foreword ... vi
Introduction ... vii
1 Scope ... 1
2 Normative references ... 1
3 Terms and definitions ... 1
4 FIX session ... 4
4.1 General ... 4
4.2 Sequence numbers ... 5
4.3 Identifying the FIX session ... 6
4.3.1 General ... 6
4.3.2 The FIX session profile ... 6
4.3.3 Identification of FIX session peers ... 6
4.3.4 Validation of SendingTime(52) ... 7
4.3.5 Additional fields available for peer identification ... 7
4.4 Establishing a FIX connection ... 7
4.4.1 Transport layer requirements ... 7
4.4.2 Using the TestMessageIndicator(464) to explicitly identify testing ... 8
4.4.3 Application layer encryption ... 8
4.4.4 Heartbeat interval ... 8
4.4.5 Heartbeat interval determination ... 8
4.4.6 Maximum message size ... 9
4.4.7 Specifying application version ... 9
4.4.8 Specifying supported message types ... 10
4.4.9 Identification of application system and FIX session processor ... 10
4.4.10 Responding to a request to establish a FIX session ... 10
4.4.11 Initial synchronization of messages in a FIX connection ... 13
4.4.12 Synchronization after successful logon ... 13
4.5 Extended features for FIX session and FIX connection initiation ... 17
4.5.1 Using NextExpectedMsgSeqNum(789) to synchronize a FIX session ... 17
4.5.2 Using ResetSeqNumFlag(141) to reset FIX session for 24 h connectivity ... 19
4.5.3 Using ResetSeqNumFlag(141) to reset FIX session during FIX connection establishment 19
4.5.4 Using initiator state to restore acceptor session state ... 19
4.6 Message exchange during a FIX connection ... 20
4.6.1 FIX connection keep alive (heartbeat) ... 21
4.6.2 Garbled message processing ... 21
4.6.3 Missing sequence number ... 21
4.6.4 Rejecting invalid messages ... 21
4.6.5 Test Request Processing ... 22
4.7 FIX connection termination ... 23
4.7.1 Normal logout processing ... 24
4.7.2 Logout without acknowledgement from peer ... 25
4.7.3 Logout with retransmission of missed messages ... 26
4.7.4 When to terminate a FIX connection by terminating the transport layer connection instead of sending a Logout(35 = 5) ... 27
4.8 Extended features for FIX connection Termination ... 27
4.8.1 Using NextExpectedMsgSeqNum(789) when terminating FIX connection due to invalid MsgSeqNum(34) ... 27
4.9 Message recovery ... 28
4.9.1 Ordered message processing ... 28
4.9.2 Request retransmission of messages ... 28
4.9.3 Responding to a ResendRequest(35 = 2) message ... 31

Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS ISO 3531-2:2022

4.9.4 Possible duplicates ... 32
4.9.5 Gap fill process ... 32
4.9.6 Sequence reset ... 34
4.9.7 Processing inbound possible duplicate messages (PossDup(43) set to "Y") ... 35
4.9.8 Processing gaps when receiving FIX session layer messages ... 35
4.10 Resending an unacknowledged application message ... 36
4.10.1 The difference between application layer resend and session layer retransmission ... 37
4.11 FIX session state matrix ... 38
4.11.1 FIX logon process state transition diagram ... 40
4.11.2 FIX logout process state transition diagram ... 41

5 FIX session profiles ... 42

5.1 FIX.4.2 session profile ... 42
5.1.1 Profile identification ... 42
5.1.2 Application version identification ... 42
5.2 FIX4 session profile ... 43
5.2.1 Profile identification ... 43
5.2.2 Application version identification ... 43
5.3 FIXT session profile ... 43
5.3.1 Profile identification ... 43
5.3.2 Multiple application version support over a single FIXT session ... 43
5.3.3 Session default application version identification ... 43
5.3.4 Message type default application version ... 44
5.3.5 Explicit application version per message ... 44
5.3.6 Use of extension packs ... 44
5.3.7 Use of a custom application version ... 45
5.4 Lightweight FIXT (LFIXT session profile) ... 45
5.4.1 Profile identification ... 46
5.4.2 Application version identification ... 46
5.4.3 LFIXT transport layer requirements ... 46
5.4.4 LFIXT compatible mode ... 46
5.4.5 LFIXT succinct mode ... 47
5.4.6 LFIXT and FIXT operating mode interoperability ... 47
5.4.7 Validation of message sequence numbers ... 48
5.4.8 Application layer recovery ... 48
5.4.9 LFIXT initiator connects to LFIXT acceptor ... 48
5.4.10 FIXT initiator connects to LFIXT acceptor (compatible mode) ... 50
5.4.11 Receipt of ResendRequest(35 = 2) message from a FIXT peer ... 53
5.4.12 Processing invalid messages ... 55

6 FIX message routing ... 55

6.1 Message routing details – one firm-to-one firm (point-to-point) ... 55
6.2 Message routing details – third party message routing ... 55

7 Transmitting alternatively encoded messages over a FIX session ... 56

7.1 Use of Attachment group ... 57

8 Components ... 57

8.1 AttachmentGrp ... 57
8.2 AttachmentKeywordGrp ... 58
8.3 HopGrp ... 58
8.4 MsgTypeGrp ... 59
8.5 StandardHeader ... 59
8.6 StandardTrailer ... 61

9 Messages ... 61

9.1 Heartbeat message ... 61
9.2 TestRequest message ... 62
9.3 ResendRequest message ... 62

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

9.4 Reject message 62
9.5 SequenceReset message 63
9.6 Logout message 63
9.7 Logon message 64
9.8 XMLnonFIX message 66

10 Fields 66

11 Code sets 72

11.1 ApplLevelRecoveryIndicatorCodeSet 72
11.2 ApplVerIDCodeSet 73
11.3 AttachmentEncodingTypeCodeSet 73
11.4 EncryptMethodCodeSet 74
11.5 GapFillFlagCodeSet 74
11.6 MsgDirectionCodeSet 74
11.7 MsgTypeCodeSet 75
11.8 PossDupFlagCodeSet 75
11.9 PossResendCodeSet 75
11.10 ResetSeqNumFlagCodeSet 76
11.11 SessionRejectReasonCodeSet 76
11.12 SessionStatusCodeSet 77
11.13 TestMessageIndicatorCodeSet 77

Bibliography 78

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

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 (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.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

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 encodings, such as the other FIX-defined encodings (e.g. FIXML, SBE, JSON, GPB, ASN.1) or non-FIX-defined encodings (e.g. XML, FpML, ISO 20022 XML, JSON).

The Financial Information eXchange session layer is used to provide reliable and recoverable messaging for electronic trading. The protocol is intended for use by asset managers, trading firms, brokerages, trading venues, clearing houses, custodians, depositories, asset servicers and others involved in the trading life cycle activities of a wide range of financial instruments. The FIX session layer functionality is a realization of the ISO/IEC 7498-1 Open System Interconnection basic reference model level 5 session layer.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

Documents
Document Sharing Hub

Full document details: BS ISO 3531-2:2022

Financial services — Financial information eXchange session layer —

Part 2:
FIX session layer

1 Scope

This document provides the normative specification of the FIX session layer standard and its session profiles.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

3.1
session layer message
message carried over the FIX session that is integral to the operation of the FIX session

3.2
application message
message that is carried over a FIX session to accomplish some business purpose

Note 1 to entry: Examples of a business purpose include an order to buy or sell a financial instrument, reporting market data or reporting an execution of a trade.

3.3
message type
identifier (code) that defines the type of message being sent

Note 1 to entry: The message type for the FIX session is a case-sensitive string encoded in the MsgType(35) field.

3.4
valid FIX message
session or application message that is a tagvalue encoded string of octets that is properly formed according to the FIX tagvalue encoding specification

3.5
FIX session processor
combination of computer hardware, firmware and software that implements the FIX session layer

Note 1 to entry: Commonly referred to as a FIX Engine.

3.6
initiator
FIX session processor that establishes the transport layer connection and initiates the session via transmission of the initial Logon(35=A) message

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

3.7
acceptor
FIX session processor that is able to establish a transport layer connection and receive Logon(35=A) requests from FIX session initiators to start or resume a FIX session

3.8
rules of engagement
specification, usually provided in document form, that describes a specific use of FIX

Note 1 to entry: Often referred to as a FIX service offering. FIX Orchestra is a standard that may be used to specify machine-readable rules of engagement.

3.9
peer
FIX session processor being communicated with over a FIX session

Note 1 to entry: The peer of the initiator is the acceptor. The peer of the acceptor is the initiator. The initiator and acceptor are peers.

3.10
counterparty
firm, legal entity or individual agreeing to use the FIX session layer to conduct some form of business endeavour

3.11
NextNumIn
Each FIX session processor must keep track of the next message sequence number it is expecting to receive from its peer to guarantee ordered delivery of messages over the life of a FIX session that may span multiple FIX connections. The next expected incoming sequence number (NextNumIn) is compared to the value in the MsgSeqNum(34) field in each message received from the peer.

3.12
NextNumOut
Each FIX session processor must keep track of the next outbound sequence number (NextNumOut) it will send to its peer over a FIX connection.

3.13
retransmission
resending of a message that was previously sent in order to resynchronize the FIX session

Note 1 to entry: A retransmitted message uses the original MsgSeqNum(34) value with PossDupFlag(43)=Y.

3.14
resend
application message being resent by an application layer because it has not received an application layer acknowledgement for the message

Note 1 to entry: A resend may only be initiated by the application layer, not the session layer. The determination of whether the message was previously received is the responsibility of the application layer, not the session layer.

3.15
gap fill
process to resolve gaps in message sequence numbers within a FIX session

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

3.16
application version
The FIX session layer provides fields to communicate versions of the application layer messages. The FIX Trading Community defines standard application versions.

3.17
extension pack
approved addition to the FIX standard

Note 1 to entry: The granularity of an extension pack may vary widely from a single enumeration value addition to the definition of entire new categories of messages. Extension packs are identified by a sequential integer number and must be applied in order. An extension pack is considered available for use if it has been approved and published by the FIX Global Technical Committee. Extension packs are created on an as-needed basis and are generally driven by community requests. An extension pack is cumulative, in that the artefacts (Orchestra file, FIXimate) include all previous extension packs. When an extension pack is published, it becomes the *FIX Latest* version of FIX.

3.18
session profile
extension or restriction on the use of the standard session layer messages that can be used to represent context of usage

Note 1 to entry: FIX4, FIXT, and LFIXT are the current FIX session profiles.

3.19
CompID
alphanumeric identifier for the entity associated with a FIX session

Note 1 to entry: As this is likely a financial markets participant company, the name was viewed colloquially as a company identifier abbreviated as a *CompID*.

3.20
SubID
subidentifier optionally available to identify a subentity within a CompID

Note 1 to entry: The SubID may be used to identify a specific trader or a subunit of a business entity. The use of SubIDs is at the discretion of the counterparties. Certain regulatory regimes globally require the use of SubID to identify specific traders.

3.21
LocationID
location identifier providing additional location information either geographical or within a trading desk on a trading floor

Note 1 to entry: The use of LocationIDs is at the discretion of the counterparties.

3.22
transport layer connection
A FIX session relies on a transport layer to provide for ordered delivery of messages and message recovery during the life of the transport layer connection. The FIX session does not require a specific transport layer, although TCP/IP is widely used and is a de facto standard transport layer for FIX sessions. TCP/IP provides an ordered reliable delivery of a stream of bytes during the life of the TCP connection. The FIX session processor reads this stream identifying FIX message boundaries.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

3.23
in-band communication
transmission of control information or metadata about the application layer or session layer in application and session FIX message types over the FIX session

Note 1 to entry: Sending the HeartBtInt(108) field in the Logon(35=A) message is an example of in-band transmission of control information. Providing version information about a FIX service in the CstmApplVerID(1129) on the Logon(35=A) message is another example.

3.24
out-of-band communication
exchange of control information or metadata about a FIX session via a separate communication mechanism than the FIX session

Note 1 to entry: Providing the TestRequestThreshold in a rules of engagement document is an example of exchanging information out-of-band. Providing version information about a FIX service on a website or in a rules of engagement document is another example of out-of-band communication. Providing access to a specification online via a website or web service would be another example of out-of-band communication. Even though this is electronic communication, it is not being done over the FIX session to which that information applies.

3.25
TestRequestThreshold
amount of time expressed as a multiplier of the heart beat interval before a TestRequest(35=1) message is sent to the peer when the heartbeat interval has been exceeded without receiving a message from the peer

Note 1 to entry: This value may be specified out-of-band in a rules of engagement.

3.26
SendingTimeThreshold
amount of time expressed in seconds in which the SendingTime(52) value in an inbound message differs from the system time available to the receiving FIX session processor

Note 1 to entry: This value may be specified out-of-band in a rules of engagement.

3.27
LogoutAckThreshold
amount of time expressed in seconds that a FIX session processor that has transmitted a Logout(35=5) request will wait for the Logout(35=5) acknowledgement before terminating the transport layer connection

Note 1 to entry: This value may be specified out-of-band in a rules of engagement.

4 FIX session

4.1 General

A **FIX session** is a bidirectional stream of ordered messages between two peers within a continuous sequence number series beginning with 1. A single FIX session can exist across multiple sequential (not concurrent) FIX connections, which means that peers may intentionally or unintentionally connect and disconnect multiple times while maintaining a single FIX session. The FIX session can be thought of as a bidirectional durable session sharing characteristics of the guaranteed delivery and durable subscriber enterprise integration architecture patterns.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

A **FIX connection** consists of three parts: logon process, message exchange (inclusive of resynchronization of state) and possible logout process over a transport layer connection. A FIX connection may be concluded by the unrecoverable loss of the transport layer, a system failure or an application failure.

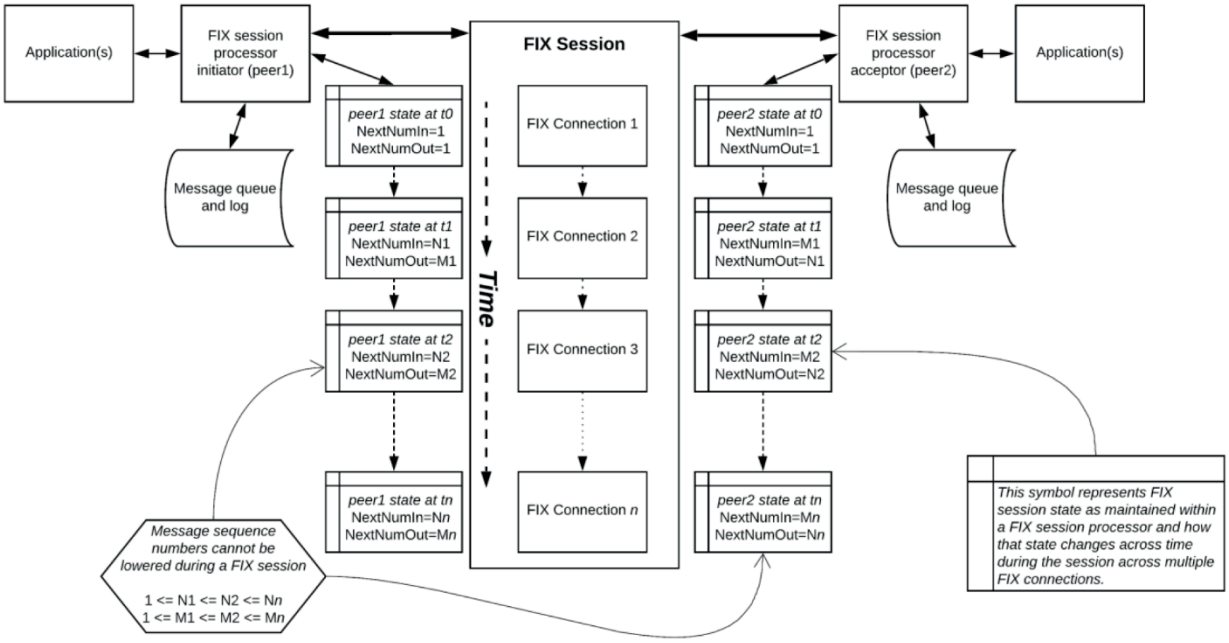

Figure 1 — Conceptual view of FIX session layer

Connecting parties shall bilaterally agree upon the time the FIX session is started and the duration of a FIX session.[1] A FIX session is often configured to correspond to a certain period of time, such as a trading day, calendar day or a trading session. A FIX session may extend beyond multiple periods by counterparty agreement.

The FIX session is based on an optimistic model. Normal delivery of data are assumed (i.e. no session layer acknowledgement of individual messages) with errors in delivery identified by message sequence number gaps.

4.2 Sequence numbers

All messages sent over a FIX session shall be identified by a unique sequence number in the MsgSeqNum(34) field.

A FIX session shall start with a next expected outgoing sequence number (NextNumOut) of 1 and a next expected incoming sequence number (NextNumIn) of 1.

A FIX session processor must maintain the NextNumOut and NextNumIn for the entire FIX session.

A FIX session processor shall persist the NextNumOut and NextNumIn in order to support FIX session recovery across multiple FIX connections.

Resetting NextNumOut to 1 and NextNumIn to 1, for whatever reason, shall constitute the beginning of a new FIX session.

FIX session layer and application layer messages shall share the same sequence number space.

[1] Firms may document the session layer configuration using the FIX Orchestra interface specification.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

Each FIX session layer and application layer message sent consumes the next outbound sequence number, incrementing NextNumOut by 1.

Each FIX session layer and application layer message received consumes the next inbound sequence number, incrementing NextNumIn by 1.

A FIX session must always have peer1.NextNumIn < = peer2.NextNumOut and peer2.NextNumIn < = peer1.NextNumOut to be considered in a valid and recoverable state.

4.3 Identifying the FIX session

4.3.1 General

A FIX session is identified by the unique combination of BeginString(8) + initiator CompID + acceptor CompID.

4.3.2 The FIX session profile

BeginString(8) shall be used to identify the FIX session profile or version of FIX.

These FIX session profiles are defined within this specification.

Table 1 — The FIX session profiles

FIX session profile	BeginString(8)	Description
FIX.4.2	FIX.4.2	The FIX session profile for use with the FIX.4.2 application layer.
FIX4	FIX.4.4	The FIX session profile backward compatible with FIX.4.4 recommended when counterparties will only be using a single application version during the FIX session, such as *FIX Latest*.
FIXT	FIXT.1.1	The FIX session profile that must be used when mixing multiple application versions over the same FIX session. May be used with a single application version of FIX such as *FIX Latest*.
LFIXT	FIXT.1.1	Lightweight FIXT restricted session layer message recovery[2] to simplify the protocol while maintaining compatibility with FIXT when using LFIXT compatible model of operation.

FIX session acceptor may be configured to support multiple FIX session profiles (or *FIX versions* prior to FIX.4.4) over the same transport layer with only one FIX session profile (or *FIX version* prior to FIX.4.4) being used per FIX session.

The FIX session acceptor should use the incoming Logon(35 = A) request BeginString(8) to identify the FIX session profile or FIX version being requested by the initiator.

4.3.3 Identification of FIX session peers

FIX relies on alphanumeric strings known as CompIDs to identify the initiator and the acceptor of FIX messages.

Counterparties must agree to their respective CompID values, which act as an identifier for the peer.

Each message sent over a FIX session layer must contain the sending entity in SenderCompID(49) and the receiving (destination) entity in TargetCompID(56). This requirement applies to both session layer and application layer messages.

[2] The LFIXT session profile uses the same BeginString(8) value as the FIXT session profile. The use of LFIXT and its mode of operation must be agreed upon out-of-band by counterparty agreement.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

FIX service processors must validate the SenderCompID(49) and TargetCompID(56) of each message and ensure that it is identical to the values present in the FIX connection Logon(35 = A) message. A discrepancy in the SenderCompID(49)+TargetCompID(56) pair should result in the termination of the FIX connection by sending a Logout(35 = A) message using the Text(58) field to indicate the error, followed by termination of the transport layer connection.

4.3.4 Validation of SendingTime(52)

SendingTime(52) must be set to the time the message is transmitted by the FIX session processor, not the time the message was queued for transmission.

SendingTime(52) must be in UTC (Universal Time Coordinated).

SendingTime(52) should be within the reasonable time of a synchronized time source of the receiving FIX session processor. This *SendingTimeThreshold* is highly dependent upon the type of application using the FIX session layer. The *SendingTimeThreshold* may be as low as seconds for high volume, low latency applications up to several minutes for certain order routing applications.

The *SendingTimeThreshold* may be defined and communicated to counterparties out-of-band within the rules of engagement.

The receiving peer shall transmit a Reject(35 = 3) message with SessionRejectReason(373) set to 10 (SendingTime Accuracy Problem) followed by a Logout(35 = 5) request when the SendingTime(52) is not within a specified tolerance of a synchronized clock.

4.3.5 Additional fields available for peer identification

The FIX session layer provides additional fields which may be used to further identify recipients within the respective counterparties.

Each counterparty may provide a subidentifier (SenderSubID(50) and TargetSubID(57)) known as SubID.

Each counterparty may provide a location identifier (SenderLocationID(142) and TargetLocationID(143)) known as LocationID.

The SubID and LocationID may vary across messages across the same FIX session.

The counterparties must agree upon the context and semantics of the SubID and LocationID if used.

4.4 Establishing a FIX connection

Establishing a FIX connection involves three distinct operations: creation of a transport layer connection, acceptance with optional authentication of the initiator by the acceptor and message synchronization (initialization).

The initiator shall establish a transport layer connection with the acceptor to establish a FIX connection.

4.4.1 Transport layer requirements

FIX session layer requires that the transport layer provides ordered and lossless message delivery and full duplex operation for the life of the transport layer connection.

FIX implementations that use the TCP/IP protocol as the transport layer must use the FIX-over-TLS (FIXS) specification which specifies the use of Transport Layer Security (TLS) with the FIX session layer to provide transport layer encryption.

The initiator shall send a Logon(35 = A) request message to initiate a FIX connection.

The Logon(35 = A) request message must always be the first message transmitted over a FIX connection. If the acceptor receives anything other than a valid Logon(35 = A) request message, an error should be logged and the transport layer connection terminated without Logout(35 = 5) processing.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

4.4.2 Using the TestMessageIndicator(464) to explicitly identify testing

Counterparties should populate the TestMessageIndicator(464) field with "Y" in the Logon(35=A) request and acknowledgement when testing.

FIX counterparties should validate that the TestMessageIndicator(464) field corresponds to the environment.

If the TestMessageIndicator(464) with value "Y" is present in a Logon(35 = A) message indicating a test session and the environment is not a test environment, the peer should transmit a Logout(35 = 5) message with the Text(58) indicating that the TestMessageIndicator(464) was set to "Y" and the environment is a production environment. Certain venues support sending test messages within the production environment. Counterparties should communicate how testing is conducted within the production environment to prevent test messages from being accepted as production messages.

If the TestMessageIndicator(464) with value "N" is present in a Logon(35 = A) message indicating a production session and the environment is a test environment, the peer should transmit a Logout(35 = 5) message with the Text(58) indicating that the TestMessageIndicator(464) was set to "N" and the environment is a test environment.

Failure to validate the TestMessageIndicator(464) against the environment may potentially lead to financial losses.

4.4.3 Application layer encryption

The use of application layer message encryption has been discouraged since the FIX.4.3 version of the FIX protocol.

EncyptionMethod(98) must be present in the Logon(35 = A) message and set to "0" when application layer encryption is not in use.

Firms may implement application layer encryption by counterparty agreement.

4.4.4 Heartbeat interval

The HeartBtInt(108) field must be present and specify the interval in seconds for generating heartbeat messages during periods of inactivity.

The HeartBtInt(108) field must be present in the Logon(35 = A) message.

The HeartBtInt(108) value must be agreed upon by the initiator and the acceptor.

The initiator and acceptor must use the same HeartBtInt(108) value within a FIX connection.

The HeartBtInt(108) value is the maximum time allowed between receipt of messages from the peer.

Each FIX session processor should implement a heartbeat interval timer. The timer should be reset upon receipt of each message from the peer.

4.4.5 Heartbeat interval determination

The acceptor should convey the rules placed on the expected heartbeat interval via out-of-band rules of engagement[3] when such rules are required by the acceptor.

There are three methods for determining the heartbeat interval:

1. Acceptor requires a specific heartbeat interval

2. Acceptor requires initiator to specify a value within a heartbeat interval range

[3] FIX Orchestra may be used to communicate which FIX session profile and which operating mode is being used for a FIX enabled service.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

3. Acceptor accepts the initiator specified heartbeat interval

4.4.5.1 Acceptor requires a specific heartbeat interval

The acceptor may refuse to establish a FIX connection if the HeartBtInt(108) value received on the Logon(35=A) message sent by the initiator does not match the heartbeat interval required by the acceptor.

The acceptor must terminate the FIX session using a Logout(35 = 5) message when the acceptor refuses to establish a FIX connection due to the inbound HeartBtInt(108) value not matching the expected value.

The acceptor should provide the reason for terminating the FIX connection in the Text(58) field of the Logout(35 = 5) message by setting it to "Invalid HeartBtInt(108), expected value N seconds", where N is the number of seconds expected in the HeartBtInt(108) field on the initiator Logon(35 = A) message. N shall be larger than zero.

4.4.5.2 Acceptor requires initiator specify a value within a heartbeat interval range

The acceptor may refuse to establish a FIX connection if the HeartBtInt(108) value received on the Logon(35=A) message sent by the initiator does not fall within a heartbeat interval range required by the acceptor.

The acceptor must terminate the FIX connection using a Logout(35 = 5) message when the acceptor refuses to establish a FIX session due to the inbound HeartBtInt(108) value not falling within the required heartbeat interval range.

The acceptor should specify the reason for terminating the connection in the Text(58) field of the Logout(35 = 5) message by setting it to "Invalid HeartBtInt(108), expected value between N and M seconds", where N and M are the start and end values of the range of the heartbeat interval in seconds permitted by the acceptor. N shall be larger than zero and M shall be larger than N.

4.4.5.3 Acceptor accepts the initiator specified heartbeat interval

The acceptor may accept the heartbeat interval specified by the initiator in the HeartBtInt(108) field of the initiator Logon(35=A) message by echoing back the heartbeat interval of the initiator in the Logon(35=0) acknowledgement.

4.4.6 Maximum message size

The initiator and acceptor may specify the maximum message size supported in the MaxMessageSize(383) field in the Logon(35=A) message.

Either peer may terminate a FIX connection due to inability to process the maximum message size specified by the other peer by sending a Logout(35 = 5) message.

The peer terminating the FIX connection should specify the reason in the Text(58) field of the Logout(35 = 5) message by setting it to "MaxMessageSize(383)=*InboundValue* exceeds maximum message size of *maximum message size*".

4.4.7 Specifying application version

Three fields are provided to identify the application version to counterparties in the Logon(35=A) message. The use of these fields varies across FIX session profiles. For instance, these fields are not applicable to the FIX.4.2 session profile as the FIX.4.2 session profile may only be used with the FIX 4.2 application version.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

Field	Definition
DefaultApplVerID(1137)	The FIX application version upon which the application layer is derived.
DegaultApplExtID(1407)	An extension pack that is used with the application layer.
DefaultCstmApplVerID(1408)	User defined custom application version information. DefaultCstmAppVerID(1400) may be used to identify the FIX Orchestra machine readable rules of engagement.

4.4.8 Specifying supported message types

The initiator and acceptor may specify the types of FIX messages by using the repeating group MsgTypeGrp in the Logon(35=A) message.

4.4.9 Identification of application system and FIX session processor[4]

By counterparty agreement, peers may identify the application system as well as the FIX session processor ("FIX Engine") in use by their respective name, version, and vendor as part of the Logon(35 = A) message in the ApplicationSystemName(1603), ApplicationSystemVersion(1604), ApplicationSystemVendor(1605), FIXEngineName(1600), FIXEngineVersion(1601), and the FIXEngineVendor(1602) fields.

4.4.10 Responding to a request to establish a FIX session

The acceptor should authenticate the identity of the initiator by examining the incoming Logon(35=A) message, referred to as Logon(35=A) request.

The Logon(35 = A) message should contain the data necessary to support the authentication method chosen by counterparty agreement.

The following fields are available for exchanging credentials for authentication:

Table 2 — Fields in the Logon(35=A) message used for authentication

Field	Use
Username(553)	Clear text userid
Password(554)	Clear text password - use is not recommended
NewPassword(925)	Clear text new password - use is not recommended
EncryptedPasswordMethod(1400)	Method of encryption used for password field
EncryptedPasswordLen(1401)	Length in octets of EncryptedPassword(1402)
EncryptedPassword(1402)	Encrypted password
EncryptedNewPasswordLen(1403)	Length in octets of EncryptedNewPassword(1403)
EncryptedNewPassword(1404)	Encrypted new password permits changing password during logon
RawDataLength(95)	Length in octets of RawData(96)
RawData(96)	Binary octet stream that can be used for authentication

SessionStatus(1409) may be used by counterparty agreement to communicate session status regarding authentication on both Logon(35 = A) and Logout(35 = 5) messages.[5]

The acceptor shall respond with a Logon(35 = A) message to accept the FIX connection initiation when the request is accepted, referred to as the Logon(35 = A) acknowledgement.

[4] Functionality added by EP113.
[5] Field added in EP56

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

The session acceptor should send a Logout(35 = 5) message to reject FIX connection initiation and provide a reason for the rejection in the Text(58) field and then immediately terminate the transport layer connection.

The initiator should not transmit any application message until the Logon(35 = A) acknowledgement has been received from the acceptor (see the next section for further recommendations).

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

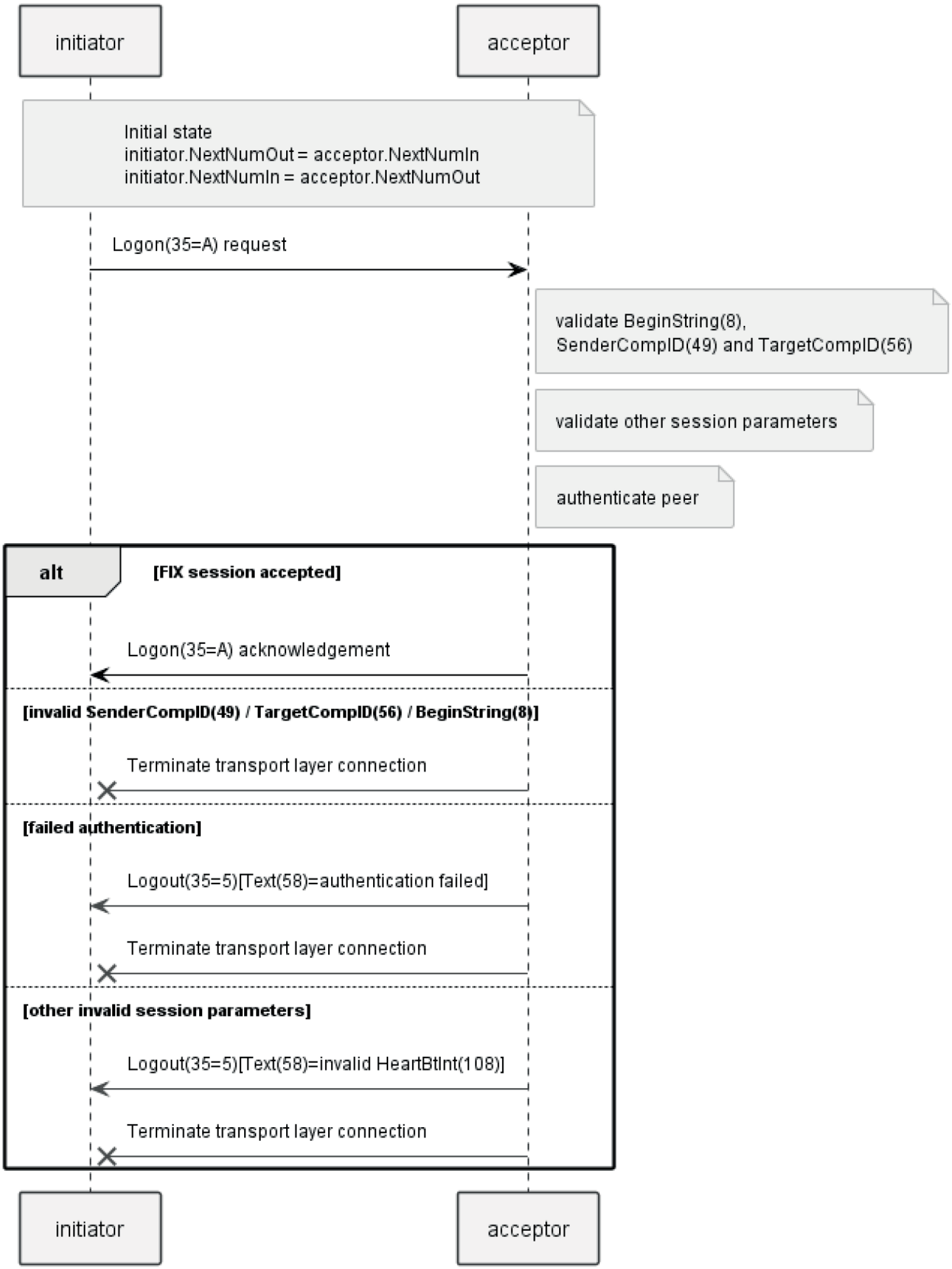

Figure 2 — FIX connection establishment

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

4.4.11 Initial synchronization of messages in a FIX connection

The acceptor shall compare MsgSeqNum(34) in the Logon(35=A) request received from the initiator with NextNumIn maintained by the acceptor.

The initiator shall compare MsgSeqNum(34) in the Logon(35 = A) acknowledgement received from the acceptor with NextNumIn maintained by the initiator.

If MsgSeqNum(34) is present in the incoming Logon(35 = A) message and equal to NextNumIn, then normal processing may commence over the FIX connection.

If MsgSeqNum(34) is present in the incoming Logon(35 = A) message and greater than NextNumIn, then message recovery must be performed.

If MsgSeqNum(34) is present in the incoming Logon(35 = A) message and is less than NextNumIn, then a Logout(35 = 5) message should be sent assuming that an error exists in the state of either the initiator or acceptor FIX session processor, followed by a termination of the transport layer connection.

The initiator and acceptor should wait a short period of time following receipt of the Logon(35 = A) message from the counterparty before transmitting queued or new application messages to permit both sides to synchronize the FIX session.

Alternatively, the initiator and acceptor may send a TestRequest(35 = 1) message and wait for the HeartBeat(35 = 0) message sent in response to the TestRequest(35 = 1) message, before sending queued or new messages, as this practice forces both sides to perform message synchronization.

4.4.12 Synchronization after successful logon

The following sequence diagrams show three scenarios.

1. acceptor requires retransmission of messages from initiator
2. initiator requires retransmission of messages from acceptor
3. both acceptor and initiator require retransmission of messages

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

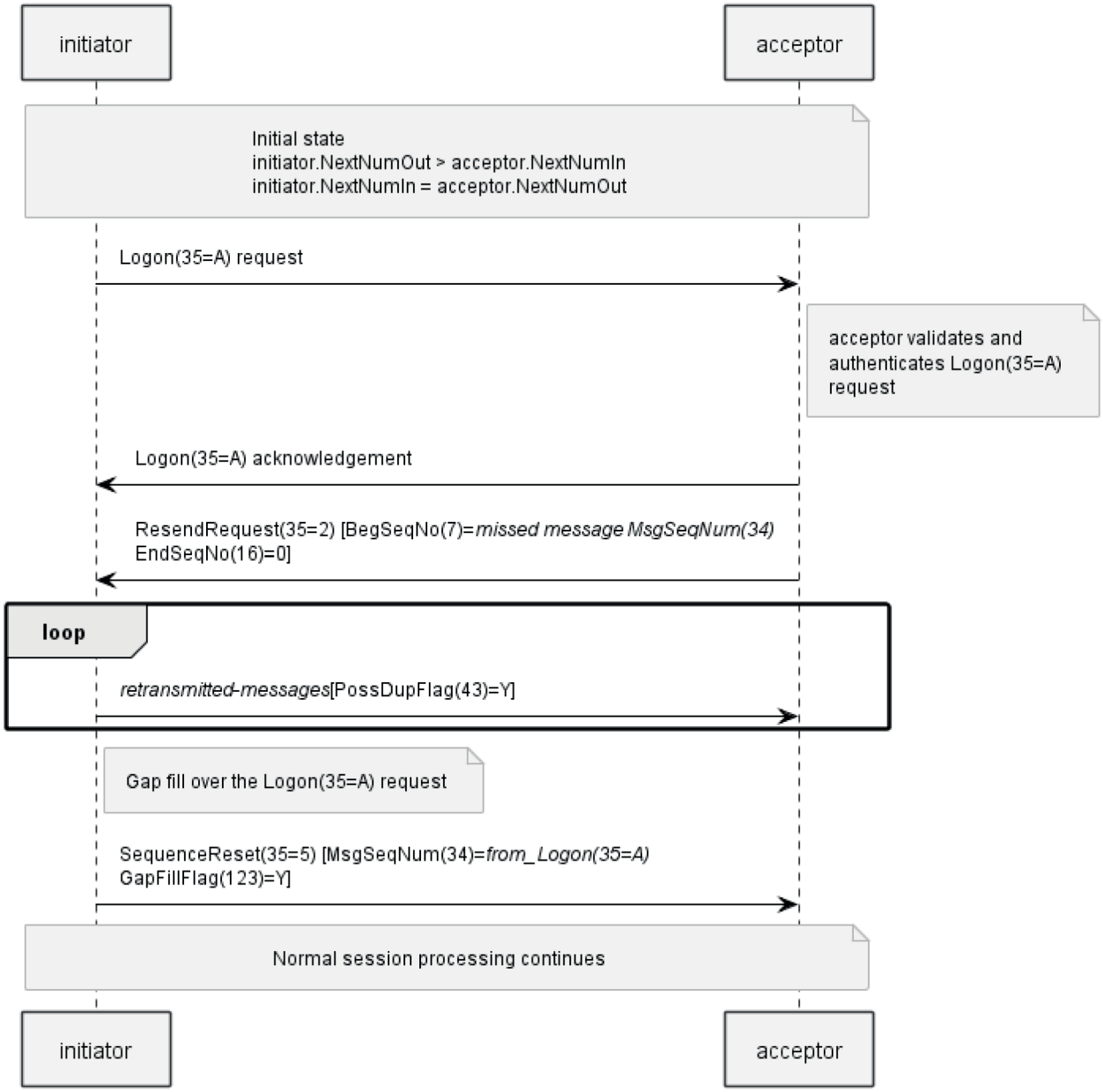

Figure 3 — ResendRequest(35=2) sent by acceptor after Logon(35=A) acknowledgement

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

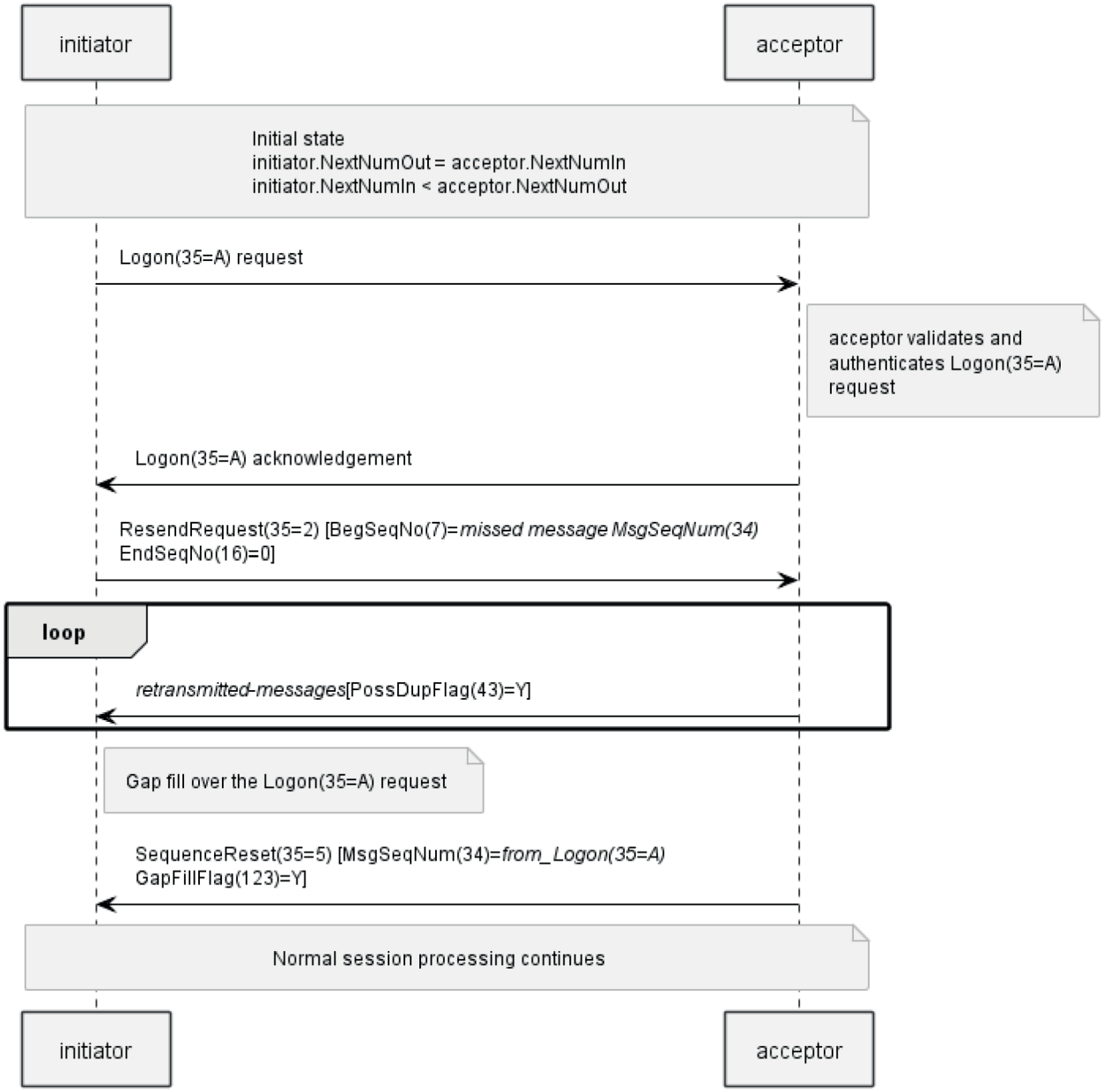

Figure 4 — ResendRequest(35=2) sent by initiator after Logon(35=A) acknowledgement

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

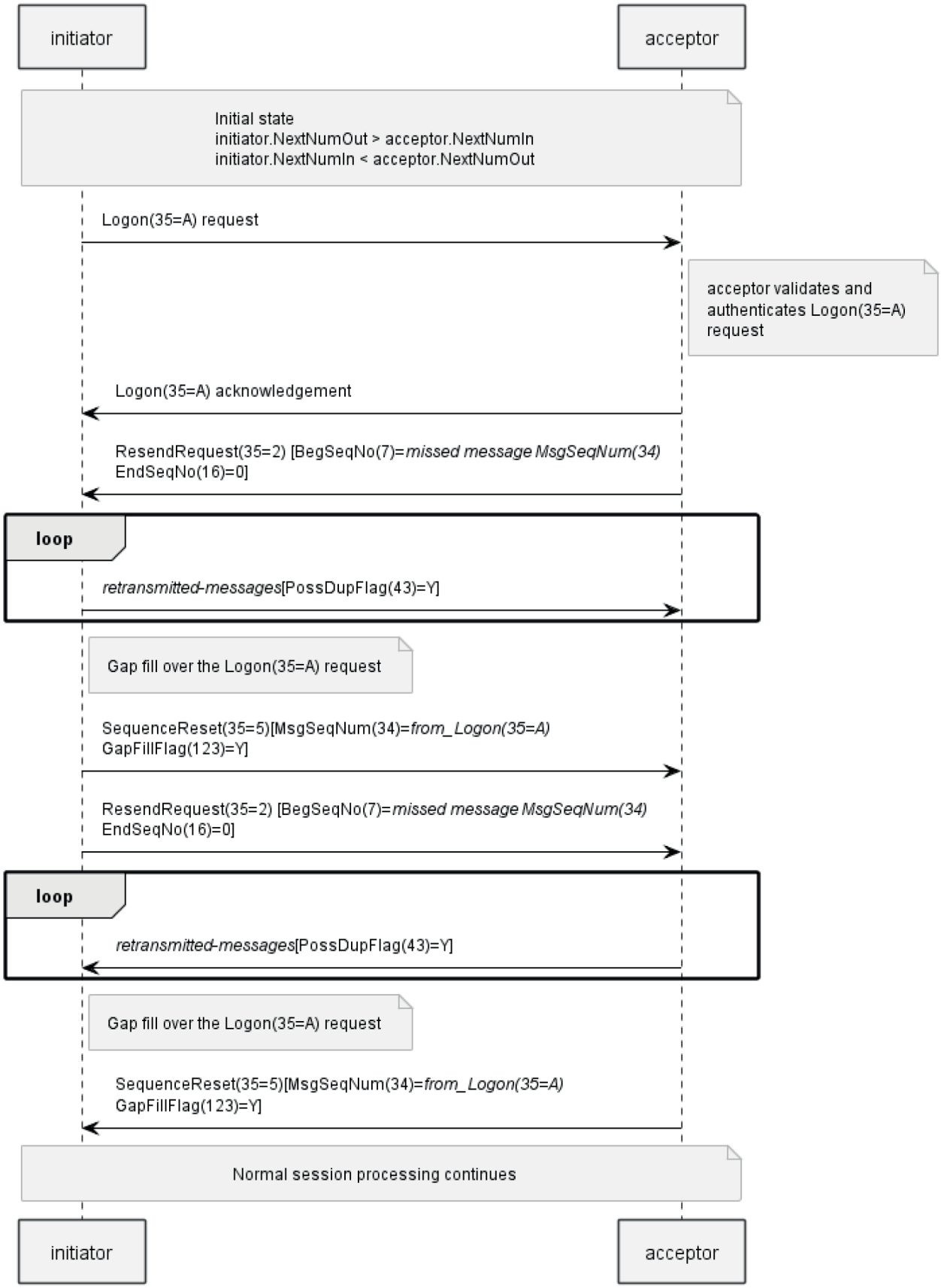

Figure 5 — ResendRequest(35=2) sent by initiator and acceptor after Logon(35=A) acknowledgement

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

4.5 Extended features for FIX session and FIX connection initiation

4.5.1 Using NextExpectedMsgSeqNum(789) to synchronize a FIX session

The FIX session processors may use the NextExpectedMsgSeqNum(789) field in the Logon(35=A) message to synchronize a FIX session by counterparty agreement defined in the FIX session rules of engagement[6].

The time required to resume a FIX session over a new FIX connection can be reduced by alerting the peer to the next message that is expected using NextExpectedMsgSeqNum(789) in the Logon(35 = A) message. Any missing messages can be retransmitted by the peer without requiring to use ResendRequest(35 = 2) messages.

A FIX session may implement next expected sequence number processing to reduce the amount of message retransmission needed to re-establish a FIX session by counterparty agreement specified in the rules of engagement.

Each FIX session processor may populate the NextExpectedMsgSeqNum(789) with the next message sequence number expected from its peer on the Logon(35 = A) when establishing a FIX connection.

Peers should not generate a ResendRequest(35 = 2) message based on MsgSeqNum(34) of the incoming Logon(35 = A) message but should expect any gaps to be filled automatically via the following process.

The receiving FIX session processor receiving a Logon(35 = A) message with NextExpectedMsgSeqNum(789) present shall compare the value of NextExpectedMsgSeqNum(789) with NextNumOut.

- If the received NextExpectedMsgSeqNum(789) is equal to NextNumOut, proceed sending new messages beginning with that number.

- If the received NextExpectedMsgSeqNum(789) is less than NextNumOut, perform message recovery for messages starting with the message with MsgSeqNum(34) equal to the NextExpectedMsgSeqNum(789) through to the message with MsgSeqNum(34) equal to NextNumOut - 1.

- If the received NextExpectedMsgSeqNum(789) is greater than NextNumOut, send a Logout(35=5) message to end the session with the Text(58) field populated explaining that "NextExpectedMsgSeqNum(789) > than last message sent".

[6] FIX Orchestra may be used to communicate which FIX session profile and which operating mode is being used for a FIX enabled service.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022

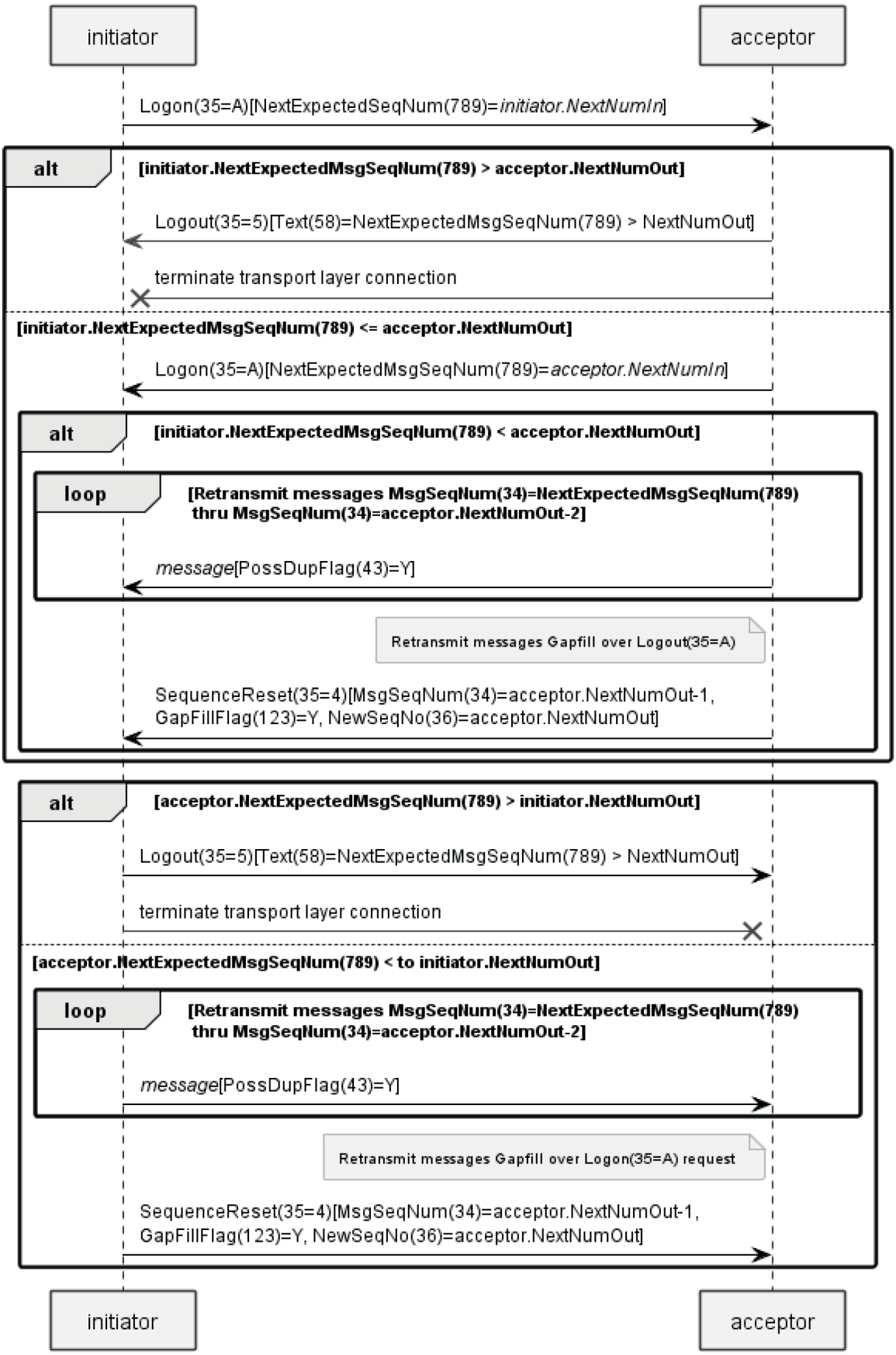

Figure 6 — Using NextExpectedMsgSeqNum(789) to synchronize the session during the logon process

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 3531-2:2022