



FIXProxy Reference Data Specification FIX50SP1 v.1.0

November 2025

Document History

Revision	Published	Summary of Changes
1.0	24.Nov.2025	First Release

Contents

1. Context.....	5
1.1 Introduction	5
1.2 Intended Audience	5
2. General Information	5
2.1 FIX User types.....	5
2.2 Supported Message Types.....	5
2.2.1 Administrative messages	5
2.2.2 Inbound Application messages	6
2.2.3 Outbound Application messages	6
2.3 Streaming Reference Data Over Fix.....	6
3. Fix Session.....	7
3.1 CompIDs	7
3.2 SenderSubID.....	7
3.3 User Authentication	7
3.4 Logon	7
3.5 Heartbeat intervals	7
3.6 Session lifetime	8
3.7 Failover and message recovery.....	8
4. Application Sequencing	8
4.1 Application Sequencing Details	8
4.1.1 Application IDs.....	8
4.1.2 The ApplicationSequenceControl group.....	9
4.1.3 The ApplIDRequestGrp.....	9
4.2 Requesting and Recovering Reference Data.....	10
4.2.1 Limitations to Reference Data recovery	10
4.2.2 Application Message Request.....	10
4.2.3 Application Message Request Ack.....	11
5. Reference Data	12
5.1 Introduction	12
5.2 Requesting Reference Data	12
5.3 Main Workflow.....	12
5.3.1 Security Definition.....	12
5.3.2 Market Definition.....	12
5.3.3 Trading Session List	12
5.3.4 Security Status	13
5.3.5 Price Reference	13
6. FIX message definitions	13
6.1 Administrative Messages.....	13
6.1.1 Logon(A).....	13

FIXProxy Reference Data SPEC FIX50SP1

6.1.2	Logout(5)	14
6.1.3	Reject(3)	14
6.1.4	Resend Request (2)	16
6.1.5	Sequence Reset (4)	16
6.1.6	Test Request (1)	16
6.1.7	Heartbeat (0)	17
6.2	Application messages	17
6.2.1	Security Definition (out)	17
6.2.2	Security Definition Request (in)	22
6.2.3	Security Definition Update Report (out)	22
6.2.4	Market Definition (out)	27
6.2.5	Trading Session List (out)	29
6.2.6	Security Status Request (in)	30
6.2.7	Security Status (out)	30
6.2.8	Price Reference Request (in)	31
6.2.9	Price Reference (out)	31
6.2.10	Application Message Request (in)	32
6.2.11	Application Message Request Ack (out)	33
7.	Component Blocks	34
7.1	Standard Header	34
7.2	Standard Trailer	35

1. CONTEXT

1.1 Introduction

This document introduces the FIX Reference Data Specification for FIX50SP1, which is dedicated to accessing AIX trading platform via FIXProxy.

1.2 Intended Audience

Technical staff of institutions which need to access AIX trading platform.

2. GENERAL INFORMATION

2.1 FIX User types

There are 3 FIX User types:

- **Order Entry** FIX user can perform order management and receives Execution reports and Trade Capture reports only for self. This user type is out of scope for this specification.
- **Drop Copy** FIX user cannot perform order management, and receives all Execution reports (except for Rejects) and Trade Capture reports for the firm. This user type is out of scope for this specification.
- **Reference Data** FIX user cannot perform order management and only receives reference data for securities.

2.2 Supported Message Types

2.2.1 Administrative messages

Logon (A)

Logout (5)

Reject (3)

Resend Request (2)

Sequence Reset (4)

Heartbeat (0)

Test Request (1)

2.2.2 Inbound Application messages

Application Message Request (BW)

Security Definition Request (c)

Security Status Request (e)

Price Reference Request (pp)

2.2.3 Outbound Application messages

Application Message Request Ack (BX)

Security Definition (d)

Security Definition Update Report (BP)

Market Definition (BU)

Trading Session List (BJ)

Security Status (f)

Price Reference (pr)

2.3 Streaming Reference Data Over Fix

AIX Reference Data over FIX is implemented using Application Sequencing.

3. FIX SESSION

The session layer conforms to the standard FIX session. Ordinary FIX message recovery is not supported. The Resend Request message is supported, but it will always be responded to with a Sequence Reset – gap fill message.

Message recovery is instead supported via application sequencing. See chapter 4 for details.

3.1 CompIDs

The Sender- and TargetCompID uniquely define the FIX session. A session can only be active (established) between two hosts simultaneously. Any attempts to establish a second FIX session using the same credentials (for instance to a backup gateway) in parallel will be rejected.

- The TargetCompID for transactions sent inbound to the Exchange will be “FIX_PROXY” for production and test systems.
- The SenderCompID for transactions sent outbound from the Exchange will be “FIX_PROXY” for production and test systems.

3.2 SenderSubID

Each inbound business transaction must have the SenderSubID (tag 50) field set to an authenticated user. One user can be authenticated by setting the Username and Password field in the Logon message.

The SenderSubID on incoming transactions will be echoed back in TargetSubID (tag 57) on outbound transactions.

3.3 User Authentication

Each incoming business transaction must have a username set in the SenderSubID field. The user must be authenticated for the transaction to be accepted. A user is authenticated by setting the Username (553) and Password (554) fields in the Logon message.

3.4 Logon

At Logon, clients are identified by:

- CompIDs
- IP Address

The Logon Username and Password fields are used to authenticate the client. When the client is authenticated, the system responds with a Logon message to the client.

3.5 Heartbeat intervals

Heartbeat intervals are negotiated at Logon using the HeartBtInt (108) field. The system allows heartbeat intervals greater than 10 seconds. Recommended heartbeat interval is 30s. A heartbeat interval set lower than 10 seconds will result in a Logout response.

3.6 Session lifetime

The FIX session lifetime is restricted to one trading day. Unlike ordinary FIX sessions the sequence number restarts at 1 after a disconnect or Logout. Message recovery using standard FIX Resend Requests is not supported. Application Sequencing is used for message recovery.

3.7 Failover and message recovery

Message recovery in this solution is based on Application Sequencing. A client must implement Application Sequencing in order to perform message recovery. For backward compatibility reasons the ordinary FIX session-level message recovery transactions are still supported. But the response to a Resend Request will be an empty Sequence Reset –gap fill message. See chapter 4 for details on Application Sequencing.

All FIX sessions have at least one primary and one secondary gateway to which the session states are fully replicated. This means that regardless to which gateway a client connects, full message recovery is provided.

A client cannot have the same FIX session active towards multiple gateway instances simultaneously.

Failover is as simple as establishing a connection to a backup gateway, and perform message recovery as described above.

NOTE: A client is not allowed to have simultaneous active connections to both a primary and a backup gateway.

4. APPLICATION SEQUENCING

4.1 Application Sequencing Details

FIX Application Sequencing is a new concept introduced in FIX 5.0. It allows for a more fine-grained subscription and recovery, where the receiver can dictate what content will be sent. As such, it is very suitable for Reference Data dissemination.

The enabling of reference data is initiated by the client, which sends an Application Message Request.

4.1.1 Application IDs

The Reference Data is separated into logical streams, called an Application. The Application is assigned a unique Application ID. Each Application is sequence numbered separately. When logged in, the client requests enabling/recovery of reference data by sequence number for each Application. Note that all Applications are sent over the same FIX session.

The Application ID used for Reference Data is:

Appl ID	Type of Data	Comment
R	Reference Data	A single Application ID for all reference data.

4.1.2 The ApplicationSequenceControl group

Every Reference Data message contains the ApplicationSequenceControl group. The group contains the Application ID and sequence number used in recovery. The receiver needs to track the sequence number for each application. ApplID (1180) contains the Application ID. Tag 1181, ApplSeqNum contains the sequence number.

ApplLastSeqNum contains the sequence number of the last sequence number sent for this Application ID on the current session. This allows gaps in the sequence. A receiver must check this field to avoid unnecessary resends.

The ApplResendFlag (1352) is only set on the responses to a request for resending of Application IDs that support full recovery. Full recovery is not available for Reference Data (see section 4.2.1 for details).

Tag no	Tag name	Comment
1180	ApplicationSequenceControl/ ApplID	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Sequence number for this Application ID.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Last sequence number for this Application ID. Used to indicate gaps in the sequence.
1352	ApplicationSequenceControl/ ApplResendFlag	Set to Y if this message is a result of a resend. Valid values: Y = Yes N = No (default if not present)

4.1.3 The ApplIDRequestGrp

The ApplIDRequestGrp is a repeating group in the Application Message Request message that contains the requested sequence numbers for each Application ID. The following fields are used:

Tag no	Tag name	Comment
1351	ApplIDRequestGrp/NoApplIDs	Number of Application IDs in this request.
1355	ApplIDRequestGrp/RefApplID	Application ID. Valid values: R = Reference Data
1182	ApplIDRequestGrp/ApplBegSeqNum	First requested sequence number. Ignored for non-recoverable Application IDs.
1183	ApplIDRequestGrp/ApplEndSeqNum	Only 0 (zero) is allowed (subscription will always be enabled).

4.2 Requesting and Recovering Reference Data

In this solution Reference Data is sent to the client after an Application Message Request has been issued.

4.2.1 Limitations to Reference Data recovery

The Application Message Request also contains the application sequence numbers which governs recovery of lost data. Note that for reference data, full recovery is not supported. A request will result in enabling the real-time transmission of the data for the Application IDs in the request. The real-time data is preceded with an initial snapshot providing the current state. Regardless of the sequence numbers given in the request, the response will always start at the next sequence number with the real-time data (preceded with a possible snapshot).

4.2.2 Application Message Request

An Application Message Request is a general request to enable reference data as well as to recover lost reference data.

The message contains a repeating group with one entry for each Application ID (see section 4.1.1 for a list of available applications). Each Application ID present in the message enables the “subscription” on that type of data (given that the user is authorized to see said data).

Request Sequence Numbers

For each Application ID it is also possible to supply a start and end application sequence number to recover lost messages.

ApplEndSeqNum (1183) is used to define the end of a set of messages to recover. If it is set to 0 it means that a subscription is enabled. In this solution a request for an Application ID always enables the subscription, ApplEndSeqNum must be set to 0.

NOTE: As a consequence of always enabling a subscription, subsequent requests for the same Application ID after Logon will result in a reject.

ApplBegSeqNum (1182) is used to set the beginning of the messages to be sent.

- For the non-recoverable Application IDs this value is ignored. A snapshot will always be sent prior to the real-time messages.
- For the fully recoverable Application IDs, this value will dictate the starting point of the recovery. If ApplBegSeqNum is higher than the last sequence number, real-time messages will be enabled without any recovery.

NOTE: The receiver must keep track of the sequence numbers on each Application ID received to be able to recover in any situation where messages have been lost.

NOTE 2: Complete message recovery for all reference data is not possible.

The Response to an Application Message Request is an Application Message Request Ack. If the request was successful, the Ack will be followed by Reference Data messages.

4.2.3 Application Message Request Ack

The Application Message Request Ack (request ack) message is the response to an Application Message Request.

The ApplResponseType field (1348) signals if the request was successful or not.

4.2.3.1. Successful Requests

For a completely successful request, the request ack will contain:

- ApplResponseType = 0 (Request successfully processed)

Following the Application Message Request Ack, the messages will always be sent in the following order:

1. Market Definitions
2. Trading Session List
3. Security Definitions
4. Security Statuses
5. Price References

NOTE: A request may fail for some Application IDs, but still be successful for other.

4.2.3.2. Requests Failing For One Or More Application Ids

If a request is made for a non-existent application id, the request ack will contain:

- ApplResponseType = 1 (application does not exist)
- ApplResponseError = 0 (application does not exist) for that Application ID.

If a request is made for an application already requested previously, the request ack will contain:

- ApplResponseType = 3 (Duplicate requests for application)
- ApplResponseError = 3 (Duplicate requests for application) for that Application ID.

If a request is made for an application with ApplEndSeqNum (1183) not set to 0 (zero), the following will be returned in the request ack:

- ApplResponseError = 1 for that Application ID.

5. REFERENCE DATA

5.1 Introduction

The systems allow the transmission of Security Definitions, Market Definitions, Security Statuses, Price References and Trading Session Lists. To enable Reference Data the receiver logs in and sends an Application Message Request.

An Application Message Request is a general request for reference data. The same message is also used to recover lost messages. A successful Application Message Request returns one or more Reference Data messages.

The System also supports requests for Security Status, Security Definition and Price Reference which will send out information for a single instrument without setting up a subscription. The unsolicited indicator (tag 325) will indicate if the Security Definition, Security Status or Price Reference message was sent out as part of a subscription or not.

5.2 Requesting Reference Data

In this solution Reference Data is sent to the client after an Application Message Request has been issued. All Reference Data messages are enabled by this single request. See chapter 4 for details.

The System also supports requests for Security Statuses, Security Definitions and Price References without setting up a subscription by sending in a Application Message Request.

5.3 Main Workflow

5.3.1 Security Definition

The Security Definition is used to publish start-of-day reference data for each tradable security in the system. For intra-day updates the Security Definition Update Report is used. Price field and corporate action updates will not trigger a Security Definition Update Message.

5.3.2 Market Definition

The Market Definition message is used to publish information on the market structure of the marketplace. Each tradable security belongs to one market (represented by one Market Definition message).

5.3.3 Trading Session List

The Trading Session List message contains all trading states (Trading Sessions) the instruments can be in. It contains information on Trading Rules, Matching Rules, and allowed order types for each state.

TradingSessionID (336) contains the actual ID of the state. Security Status messages also contain this ID to identify the state it refers to. This is in contrast to standard FIX where TradingSessionID contains enums such as DAY, HALFDAY etc.

5.3.4 Security Status

All session State change information is sent on the instrument level using the Security Status message. The Security Status message also contains information about last traded price and corporate actions.

5.3.5 Price Reference

Price reference contains reference price, current upper and lower limits, base price and previous day closing price for an instrument.

For price limits both the static and current price limits will be considered when this message is sent.

The disseminated range will be the smallest allowed range.

Example:

The static price limits are: 10.00 – 10.50

The dynamic price limits are: 10.10 – 10.60

Result: The disseminated price limits are: 10.10 – 10.50

When there is fixed matching there is no allowed price limit range. In such a case the upper and lower limit values sent in fix will be the same as the base price. When there is no price limits the upper and lower price limit tags will not be sent at all.

6. FIX MESSAGE DEFINITIONS

6.1 Administrative Messages

6.1.1 Logon(A)

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
Standard Header		Y	MsgType = A	
98	EncryptMethod	Y	(Always unencrypted)	Int
108	HeartBtInt	Y	Note same value used by both sides	Int
141	ResetSeqNumFlag	N	Indicates both sides of a FIX session should reset sequence numbers	Boolean
553	Username	Y	The FIX connector username	String
554	Password	Y	The FIX connector password. No security exists	String

			without transport level encryption	
1137	DefaultApplVerID	Y	Specifies the service pack release being applied by default to the messages in this session. The only valid value is '8' = FIX50SP1	String
Standard Trailer Y		Y		

Valid range for HeartBtInt is from 15 to 60 seconds.

If the HeartBtInt is out of this range, the server will reply with the last valid value or the default value (30).

6.1.2 Logout(5)

The logout message initiates or confirms the termination of a FIX session. Disconnection without the exchange of logout messages should be interpreted as an abnormal condition.

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
Standard Header		Y	MsgType = 5	
58	Text	N	Free format text string	String
Standard Trailer		Y		

6.1.3 Reject(3)

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation. An example of when a reject may be appropriate would be the receipt of a message with invalid basic data (e.g. MsgType=&) which successfully passes check sum and body length checks. As a rule, messages should be forwarded to the trading application for business level rejections whenever possible.

Rejected messages should be logged and the incoming sequence number incremented.

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
-----	------------	----------	----------	--------

FIXProxy Reference Data SPEC FIX50SP1

Standard Header		Y	MsgType = 3	
45	RefSeqNum	Y	MsgSeqNum of rejected message SeqNum	SeqNum
371	RefTagID	N	The tag number of the FIX field being referenced	Int
372	RefMsgType	N	The MsgType of the FIX message being referenced	String
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message. 0 – Invalid tag number 1 – Required tag missing 2 – Tag not defined for this message type 3 – Undefined tag 4 – Tag specified without a value 5 – Value is incorrect (out of range) for this tag 6 – Incorrect data format for value 9 – CompID problem 10 = SendingTime Accuracy Problem 11 = Invalid MsgType 13 = Tag appears more than once 14 = Tag specified out of required order 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 18 = Invalid/unsupported application version 99 – Other.	Int
58	Text	N	Free format text string	String
Standard Trailer		Y		

6.1.4 Resend Request (2)

Resend Request is not supported for FIX Reference Data.

6.1.5 Sequence Reset (4)

This message has two uses. The common usage is with GapFillFlag set to 'Y', which is used in a response to a Resend Request to indicate that a range of messages will not be resent. This is commonly used to avoid resending administrative messages like Heartbeats.

The other (very rare) usage is to reset the sequence numbers to a higher number to get out of a deadlock. This is only triggered by manual intervention.

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
Standard Header		Y	MsgType = 4	
123	GapFillFlag	N		Boolean
36	NewSeqNo	Y		SeqNum
Standard Trailer		Y		

6.1.6 Test Request (1)

The test request message forces a heartbeat from the opposing application. The test request message checks sequence numbers or verifies communication line status. The opposite application responds to the Test Request with a Heartbeat containing the TestReqID.

The TestReqID verifies that the opposite application is generating the heartbeat as the result of Test Request and not a normal timeout. The opposite application includes the TestReqID in the resulting Heartbeat. Any string can be used as the TestReqID (one suggestion is to use a timestamp string).

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
Standard Header		Y	MsgType = 1	
112	TestReqID	Y		String
Standard Trailer		Y		

6.1.7 Heartbeat (0)

The Heartbeat monitors the status of the communication link and identifies when the last of a string of messages was not received.

When either end of a FIX connection has not sent any data for [HeartBtInt] seconds, it will transmit a Heartbeat message. When either end of the connection has not received any data for (HeartBtInt + (HeartBtInt ÷ 2)) seconds, it will transmit a Test Request message. If there is still no heartbeat message received after (HeartBtInt + (HeartBtInt ÷ 2))seconds then the connection should be considered lost and corrective action be initiated. If HeartBtInt is set to zero then no regular heartbeat messages will be generated. Note that a test request message can still be sent independent of the value of the HeartBtInt, which will force a Heartbeat message.

Heartbeats issued as the result of Test Request must contain the TestReqID transmitted in the Test Request message. This is useful to verify that the Heartbeat is the result of the Test Request and not as the result of a regular timeout.

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
Standard Header		Y	MsgType = 0	
112	TestReqID	N	Required when the heartbeat is the result of a Test Request message.	String
Standard Trailer		Y		

6.2 Application messages

6.2.1 Security Definition (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = d
1180	ApplicationSequenceControl/ ApplID	Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name		Req'd	Comment
				sequence gaps where it cannot be avoided.
55	Symbol		Q	Short name.
107	SecurityDesc			Long name
48	SecurityID		Q	Orderbook ID
22	SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
167	SecurityType			Valid values: 1 = Options 2 = Forward 3 = Futures 4 = FRA 5 = Cash 6 = Payment 7 = Exchange Rate 8 = Interest rate swap 9 = REPO 11 = Combination 12 = Guarantee 13 = OTC general 14 = Equity warrant 15 = Security lending 17 = Strip 18 = Certificates
541	MaturityDate			Date of Maturity.
20041	NoCouponBlock			Number of listed coupons
→	223	CouponRate		
→	224	CouponPaymentDate		
306	UnderlyingIssuer			
231	ContractMultiplier			Specifies the ratio or multiply factor to convert from “nominal” units (e.g. contracts) to total units (e.g. shares).
200	MaturityMonthYear			Specifies the month and year of maturity. Format: YYYYMM
202	StrikePrice			Strike Price
15	Currency			Currency
20035	LastTradingDate			Date of Last Trading
20037	FirstTradingDate			Date of First Trading
20036	LastTradingTime			Time of Last Trading
20038	FirstTradingTime			Time of First Trading
20039	SectorCode			Sector Code
20040	SectorCodeDesc			Sector Code Description
225	IssueDate			
873	DatedDate			
454	NoSecurityAltID			Number of alternate Security Identifiers
	455	SecurityAltID		International Securities Identification Number (ISIN)
	456	SecurityAltIDSource		Valid values: 4 = ISIN number
1310	NoMarketSegments		Q	Number of market segments on which the security is traded. Will always be 1.

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name		Req'd	Comment
→	1301	MarketID	Q	Name of the market
→	1300	MarketSegmentID		Market Segment Code
→	1396	MarketSegmentDesc		Market Segment Name
→	1205	NoTickRules		Number of Tick Rules NOTE: Tick Rules are associated with the security, not with the market.
→	→	1206	StartTickPriceRange	Starting price range for specified tick increment
→	→	1207	EndTickPriceRange	Ending price range for the specified tick increment
→	→	1208	TickIncrement	Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded
	1234	NoLotTypeRules		Number of Lot Types NOTE: Lot Types are associated with the security, not with the market.
→	→	1093	LotType	Lot Type. Valid values: 2 = Round Lot
→	→	1231	MinLotSize	Lot size of lot type specified in LotType(1093). To enter an order for this particular Lot Type MatchIncrement needs to be set to this value. Note that order quantity must be a multiple of this value.
→	→	21010	LotStartPrice	
→	→	21012	LotMaxTradeVol	The maximum order quantity that can be submitted for a security. It is valid only for Normal Limit Price Orders.
→	→	21013	LotMinQty	The minimum quantity of an order for the defined price level
→	→	21068	LotMinQuote	The minimum quote quantity that can be submitted for a security.
→	→	21069	LotMaxQuote	The maximum quote quantity that can be submitted for a security.
→	→	21009	LotDisplayHighQty	The maximum show quantity allowed in an order for the defined price level
→	→	21014	LotDisplayLowQty	The minimum show quantity allowed in an order for the defined price level
→	→	21011	LotDefaultQtyLimit	Default limit in case no price source series or closing price can be found
228	Factor			Price quotation factor
1150	TradingReferencePrice			Reference price.
21003	BasePrice			Base price.
21004	TradingMethod			The two character information field from the Instrument Class.
21005	GrossSettlement			Y if gross settlement is applied on the Instrument Class level, otherwise N.
916	StartDate			For Debt market series, -Value1 date For Equity market series,

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name	Req'd	Comment
			-First trading date for extra-ordinary series. None if no date is defined.
917	EndDate		For Debt market series, -Value2 date For Equity market series, Last trading date for extra-ordinary series. None if no date is defined.
325	Unsolicited Indicator		Indicates whether the message is sent as a result of a subscription request or not. Will be 'N' in initial startup request to FIX client. Otherwise 'Y'.
21007	TradingSessionName		Name of the Trading Session
21008	PartitionId	Q	Which partition this security is traded on. Always 1 for AIX.
21026	DerivativeLevel		Specifies the level of derivation for the instrument. Valid values: 0 = Spot
21027	NumberOfSharesIssued		Specifies the total number of shares issued for this security.
21028	PhysicalDelivery		Specifies whether the instrument is defined with physical delivery or not. Valid values: Y/N
21029	Multiplier		This is an optional field that provides additional information for Warrants/Certificates.
21030	Reference		This is an optional field that provides additional information for Warrants/Certificates.
21032	InstrumentType	Q	Instrument Type
21031	SeriesDesc	Q	Valid values: S – Standard Instrument
965	SecurityStatus	Q	Valid values: 1 = Active 2 = Inactive
1148	LowLimitPrice		Current lower limit for price limits.
1149	HighLimitPrice		Current upper limit for price limits.
5011	OrderMaxValue		Maximum order value, only applicable for equity market.
38	OrderQty		
159	AccruedInterestAmt		Amount of Accrued Interest
1948	CouponFrequencyPeriod		Time unit multiplier for the frequency of fixed income instruments coupon payment.
1949	CouponFrequencyUnit		Time unit associated with the frequency of fixed income instruments coupon payment.
1950	CouponDayCount		The day count convention used in interest calculations for fixed income instruments. Valid values: 1-US30360

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name		Req'd	Comment
				4- EU30360 6- ACT360 7- ACT365 8- ACTAFB 9- ACTACT ISMA 17- ACT364
21059	CouponInterest			The coupon interest for the underlying. (Only applicable for interest-related fixed-income instruments.)
21055	OddFirstCoupon			Y if odd-first coupon is used, otherwise N.
21056	OddLastCoupon			Y if odd-last coupon is used, otherwise N.
21057	PremiumUnit			Order entry price unit Valid values: 1 – Price 2 – Yield 3 – Point 7 – Basis point 8 – Clean price 9 – Dirty price 12 – Rate
21058	FixedIncomeType			Type of fixed-income underlying. Valid values: 12 – Type 1 Discounted securities 13 – Type 2 Fixed rate bonds 14 – Type 3A Variable rate bonds next coupon is known 15 – Type 4 Index linked strips 16 – Type 5 Index linked bonds 17 – Type 6 Euro bonds USD 18 – Type 7 Euro bonds EUR 19 – Type 8A Irregular Clean price 20 – Type 3B Variable rate bonds next coupon is unknown 21 – Type 8B Irregular Dirty price
21050	NoTradeReports			Number of Trade Reports.
→	21051	TradeReportCode		Number to be used in tag 828 TrdType in Trade Reports. Please refer to the list of supported Trade Type values supplied by the marketplace.
→	21052	MinTradeRepQty		Minimum quantity in Trade Report.
→	21053	MaxTradeRepQty		Maximum quantity in Trade Report.
→	21064	TrLotsize		Trade Report Lotsize
60	TransactTime			Dissemination time = Current time
21063	DaysToMaturity / Repo Term			Number of days to maturity date
8000	ShortSaleRestriction			Specify what Short Sell validation rule that should be used. Valid values; 0 = No restrictions (Short sell is allowed with no price validation) 1 = Security is not shortable (Short sell is not allowed.) 2 = Security not shortable at or below the

Tag	FIX tag name	Req'd	Comment
			best bid (Short sell is allowed with Uptick rule price validation)
21065	IndexClassification		Specify a valid Index Classification
21066	ValueDate		Value date for underlying.
21067	ResetDate		
743	DeliveryDate		Delivery Start Date
21072	NumberOfDecimalInPrice		Number of decimals used in price for the order book.
21073	InstrumentClass		The Instrument class identity.
21074	FaceValue		Face value
	Standard Trailer	Y	

6.2.2 Security Definition Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = c
320	SecurityReqID	Y	Client-generated identifier.
321	SecurityRequestType	Y	Type of Security Definition request. Valid values: 4 = Symbol
55	Symbol	Q	Short name.
	Standard Trailer	Y	

6.2.3 Security Definition Update Report (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = BP
1180	ApplicationSequenceControl/ ApplID	Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with sequence gaps where it cannot be avoided.
980	SecurityUpdateAction	Q	Valid values: A = Add D = Delete M = Modify
55	Symbol	Q	Short name
107	SecurityDesc		Long name
48	SecurityID	Q	Orderbook ID
22	SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
167	SecurityType		Valid values: 1 = Options

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name		Req'd	Comment
				2 = Forward 3 = Futures 4 = FRA 5 = Cash 6 = Payment 7 = Exchange Rate 8 = Interest rate swap 9 = REPO 11 = Combination 12 = Guarantee 13 = OTC general 14 = Equity warrant 15 = Security lending 17 = Strip 18 = Certificates
541	MaturityDate			Date of Maturity.
20041	NoCouponBlock			Number of listed coupons
	22 3	CouponRate		
	22 4	CouponPaymentDate		
306	UnderlyingIssuer			
231	ContractMultiplier			Specifies the ratio or multiply factor to convert from “nominal” units (e.g. contracts) to total units (e.g. shares).
200	MaturityMonthYear			Specifies the month and year of maturity. Format: YYYYMM
201	PutOrCall			Indicates whether an option contract is a put or call. Valid values: 0 = Put 1 = Call
202	StrikePrice			Strike Price
15	Currency			Currency
21001	InstClassification			Instrument classification grouping
20035	LastTradingDate			Date of Last Trading
20036	LastTradingTime			Time of Last Trading
20037	FirstTradingDate			Date of First Trading
20038	FirstTradingTime			Time of First Trading
20039	SectorCode			Sector Code
20040	SectorCodeDesc			Sector Code Description
225	IssueDate			
873	DatedDate			
454	NoSecurityAltID			Number of alternate Security Identifiers
	45 5	SecurityAltID		International Securities Identification Number (ISIN)
	45 6	SecurityAltIDSource		Valid Values: 4 = ISIN number
1310	NoMarketSegments		Q	Number of market segments on which the security is traded. Will always be 1.
→	1301	MarketID	Q	Name of the Market

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name		Req'd	Comment
→	1300	MarketSegmentID		Market Segment Code.
→	1396	MarketSegmentDesc		Market Segment Name
→	1205	NoTickRules		Number of Tick Rules NOTE: Tick Rules are associated with the security, not with the market.
→	→	1206	StartTickPriceRange	Starting price range for specified tick increment
→	→	1207	EndTickPriceRange	Ending price range for the specified tick increment
→	→	1208	TickIncrement	Tick increment for stated price range. Specifies the valid price increments at which a security can be quoted and traded
→	1234	NoLotTypeRules		Number of Lot Types NOTE: Lot Types are associated with the security, not with the market.
→	→	1093	LotType	Lot Type. Valid values: 2 = Round Lot
→	→	1231	MinLotSize	Lot size of lot type specified in LotType(1093). To enter an order for this particular Lot Type MatchIncrement needs to be set to this value. Note that order quantity must be a multiple of this value.
→	→	21010	LotStartPrice	
→	→	21012	LotMaxTradeVol	The maximum order quantity that can be submitted for a security. It is valid only for Normal Limit Price Orders.
→	→	21013	LotMinQty	The minimum quantity of an order for the defined price level
→	→	21068	LotMinQuote	The minimum quote quantity that can be submitted for a security.
→	→	21069	LotMaxQuote	The maximum quote quantity that can be submitted for a security.
→	→	21009	LotDisplayHighQty	The maximum show quantity allowed in an order for the defined price level
→	→	21014	LotDisplayLowQty	The minimum show quantity allowed in an order for the defined price level
→	→	21011	LotDefaultQtyLimit	Default limit in case no price source series or closing price can be found
228	Factor			Price quotation factor
292	CorporateAction			Corporate Actions currently active for this security. Valid values: 01 = R.H. Kullandırılarak Bedelli Sermaye Art 02 = R.H. Kullandırılmadan Bedelli Sermaye Art 03 = Bedelsiz Sermaye Artırımı 04 = Sermaye Azaltımı 05 = Birlesme / Devralma 06 = Temettu

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name	Req'd	Comment
			99 = Diger Important Note: When multiple corporate actions occur, tag 292 is sent with a space character between multiple values. (i.e. "292=01 03")
1150	TradingReferencePrice		Reference price.
21003	BasePrice		Base price.
21004	TradingMethod		The two character information field from the Instrument Class.
21005	GrossSettlement		Y if gross settlement is applied on the Instrument Class level, otherwise N.
916	StartDate		For debt market series, -Value1 date For equity market series, -First trading date for extra-ordinary series. None if no date is defined..
917	EndDate		For debt market series, -Value2 date For equity market series, -First trading date for extra-ordinary series. None if no date is defined.
21007	TradingSessionName		Name of the trading session.
21008	PartitionId		Which partition this security is traded on. Always 1 for AIX
21026	DerivativeLevel		Specifies the level of derivation for the instrument. Valid values: 0 = Spot
21027	NumberOfSharesIssued		Specifies the total number of shares issued for this security.
21028	PhysicalDelivery		Specifies whether the instrument is defined with physical delivery or not. Valid values: Y/N
21029	Multiplier		This is an optional field that provides additional information for Warrants/Certificates.
21030	Reference		This is an optional field that provides additional information for Warrants/Certificates.
21032	InstrumentType	Q	Instrument Type
21031	SeriesDesc	Q	Valid values: S – Standard Instrument
965	SecurityStatus	Q	Valid values: 1 = Active 2 = Inactive
1148	LowLimitPrice		Current lower limit for price limits.

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name	Req'd	Comment
1149	HighLimitPrice		Current upper limit for price limits.
5011	OrderMaxValue		Maximum order value, only applicable for equity market.
38	OrderQty		
159	AccuredInterestAmt		Amount of Accrued Interest
1948	CouponFrequencyPeriod		Time unit multiplier for the frequency of fixed income instruments coupon payment.
1949	CouponFrequencyUnit		Time unit associated with the frequency of fixed income instruments coupon payment.
1950	CouponDayCount		The day count convention used in interest calculations for fixed income instruments. Valid values: 1-US30360 4-EU30360 6- ACT360 7- ACT365 8- ACTAFB 9- ACTACT ISMA 17- ACT364
21059	CouponInterest		The coupon interest for the underlying. (Only applicable for interest-related fixed-income instruments.)
21055	OddFirstCoupon		Y if odd-first coupon is used, otherwise N.
21056	OddLastCoupon		Y if odd-last coupon is used, otherwise N.
21057	PremiumUnit		Order entry price unit Valid values: 1 - Price 2 – Yield 3 - Point 7 - Basis point 8 – Clean price 9 – Dirty price 12 – Rate
21058	FixedIncomeType		Type of fixed-income underlying. Valid values: 12 – Type 1 Discounted securities 13 – Type 2 Fixed rate bonds 14 – Type 3A Variable rate bonds next coupon is known 15 – Type 4 Index linked strips 16 – Type 5 Index linked bonds 17 – Type 6 Euro bonds USD 18 – Type 7 Euro bonds EUR 19 – Type 8A Irregular Clean price 20 – Type 3B Variable rate bonds next coupon is unknown 21 – Type 8B Irregular Dirty price

Tag	FIX tag name		Req'd	Comment
21050	NoTradeReports			Number of Trade Reports.
→	21051	TradeReportCode		Number to be used in tag 828 TrdType in Trade Reports. Please refer to the list of supported Trade Type values supplied by the marketplace.
→	21052	MinTradeRepQty		Minimum quantity in Trade Report.
→	21053	MaxTradeRepQty		Maximum quantity in Trade Report.
→	21064	TrLotsize		Trade Report Lotsize
60	TransactTime			Dissemination time = Current time
21063	DaysToMaturity / Repo Term			Number of days to maturity date
8000	ShortSaleRestriction			Specify what Short Sell validation rule that should be used. Valid values; 0 = No restrictions (Short sell is allowed with no price validation) 1 = Security is not shortable (Short sell is not allowed.) 2 = Security not shortable at or below the best bid (Short sell is allowed with Uptick rule price validation)
21065	IndexClassification			Specify a valid Index Classification
21066	ValueDate			Value date for underlying.
21067	ResetDate			
743	DeliveryDate			Delivery Start Date
21072	NumberOfDecimalsInPrice			Number of decimals used in price for the order book.
21073	InstrumentClass			The Instrument class identity.
21074	FaceValue			Facevalue
	Standard Trailer		Y	

6.2.4 Market Definition (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = BU
1180	ApplicationSequenceControl/ ApplID	Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with sequence gaps where it cannot be avoided.
1394	MarketReportID	Y	Required in FIX. Will be set, but can be ignored.
1301	MarketID	Y	Market identifier.

FIXProxy Reference Data SPEC FIX50SP1

Tag	FIX tag name	Req'd	Comment
	Standard Trailer	Y	

6.2.5 Trading Session List (out)

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = BJ
1180	ApplicationSequenceControl/ ApplID		Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum		Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum		Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with sequence gaps where it cannot be avoided.
386	NoTradingSessions		Y	Number of Trading Sessions (states) listed in this message
→	336	TradingSessionID	Y	ID of Trading Session
→	1326	TradingSessionDesc	Q	Human-readable name of Trading Session
→	1237	NoOrdTypeRules		Will always be 1
→	→	40	OrdType	Shows whether Market orders are allowed in this state. Valid values: 1 = Market
→	1239	NoTimeInForceRules		Indicates number of allowed distinct order type rules
→	→	59	TimeInForce	Shows whether IOC or FOK orders are allowed in this state. Valid values: 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FOK)
→	1235	NoMatchRules		Will always be 1
→	→	1142	MatchAlgorithm	Required in FIX if group is present. Always set to [N/A]
→	→	574	MatchType	Valid values: 4 = Auto-match
→	21024	OffHoursTrading		Default value. N
	Standard Trailer		Y	

6.2.6 Security Status Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = e
263	Subscription Request Type	Q	Subscription request type. Valid values: 0 = Snapshot
55	Symbol	Q	Short name.
324	SecurityStatusReqID	Q	Client-generated identifier.
	Standard Trailer	Y	

6.2.7 Security Status (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = f
1180	ApplicationSequenceControl/ ApplID	Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with sequence gaps where it cannot be avoided.
55	Symbol	Q	Short name.
48	SecurityID	Q	Orderbook ID
22	SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
336	TradingSessionID		ID of Session State
326	SecurityTradingStatus		Will be sent out when a security is suspended or resumed. Values: 2 = Trading Halt 3 = Resume
325	Unsolicited Indicator		Indicates whether the message is sent as a result of a subscription request or not. Will be 'N' in initial startup request to FIX client. Otherwise 'Y'.
31	LastPX		Last trade price in the related security in the last session. This price is not updated in real time after every trade.
	Standard Trailer	Y	

6.2.8 Price Reference Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = pp
55	Symbol	Q	Short name.
	Standard Trailer	Y	

6.2.9 Price Reference (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = pr
1180	ApplicationSequenceControl/ ApplID	Q	Application ID. Valid values: R = Reference Data
1181	ApplicationSequenceControl/ ApplSeqNum	Q	Application sequence number assigned to the message by the application generating the message.
1350	ApplicationSequenceControl/ ApplLastSeqNum	Q	The previous sequence number in the application sequence stream. Permits an application to publish messages with sequence gaps where it cannot be avoided.
55	Symbol	Q	Short name.
48	SecurityID	Q	Orderbook ID
22	SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
325	Unsolicited Indicator	Q	Indicates whether the message is sent as a result of a subscription request or not. Will be 'N' in initial startup request to FIX client. Otherwise 'Y'.
1148	LowLimitPrice		Current lower limit for price limits.
1149	HighLimitPrice		Current upper limit for price limits.
1150	TradingReferencePrice		Reference Price
21003	Base Price		Base Price
21025	TheoreticalPrice		Theoretical Price
140	PrevClosePx		Previous day's closing price
60	TransactTime		Dissemination time = Current time
	Standard Trailer	Y	

6.2.10 Application Message Request (in)

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = BW
1346	ApplReqID		Y	Unique identifier for request
1347	ApplReqType		Y	Type of Application Message Request being made. Valid values: 1 = Subscription to the specified Applications
1351	NoApplIDs			Number of Application IDs in this request
→	1355	RefApplID	Q	Application ID requested. Valid values: R = Reference Data
→	1182	ApplBegSeqNum		Sequence number of first message to be resent.
→	1183	ApplEndSeqNum		Last Sequence number of message to be resent or 0 (zero) for all messages.
	Standard Trailer		Y	

6.2.11 Application Message Request Ack (out)

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = BX
1353	ApplResponseID		Y	Identifier for the Application Message Request Ack
1346	ApplReqID		Q	Identifier of the request associated with this ACK message
1347	ApplReqType		Q	Type of Application Message Request being made. Valid values: 1 = Subscription to the specified Applications
1348	ApplResponseType		Q	Used to indicate the type of acknowledgement being sent. Valid Values: 0 = Request successfully processed 1 = Application does not exist 2 = Messages not available 3 = Duplicate requests for application
1351	NoApplIDs			Number of Application IDs in this request
→	1355	RefApplID		Application ID requested. Valid values: R = Reference Data
→	1182	ApplBegSeqNum		Sequence number of first message to be resent.
→	1183	ApplEndSeqNum		Last Sequence number of message to be resent or 0 (zero) for all messages.
→	1354	ApplResponseError		Valid values: 0 = Application does not exist 1 = Messages requested are not available 3 = Duplicate requests for application
58	Text			
	Standard Trailer		Y	

7. COMPONENT BLOCKS

7.1 Standard Header

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
8	BeginString	Y	Must be first field in message, the value must be "FIXT.1.1"	String
9	BodyLength	Y	Must be second field in message.	Length
35	MsgType	Y	Must be third field in message.	String
49	SenderCompID	Y	Identifies the firm sending the message.	String
56	TargetCompID	Y	Identifies the firm receiving the message.	String
34	MsgSeqNum	Y	Sequence Number	SeqNum
50	SenderSubID	N	Assigned value used to identify specific message originator (e.g. desk, trader, etc.)	String
57	TargetSubID	N	Assigned value used to identify specific message receiver (e.g. desk, trader, etc.)	String
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.	Boolean
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number.	Boolean

FIXProxy Reference Data SPEC FIX50SP1

52	SendingTime	Y	Can be embedded within encrypted data section.	UTCTimeStamp
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime	UTCTimeStamp

7.2 Standard Trailer

TAG	FIELD NAME	REQUIRED	COMMENTS	FORMAT
10	Checksum	Y	Always last field in message.	String