



dxFeed Market Data: FIX Protocol Specification

April 2018

Information in this document is subject to change without notice and does not represent a commitment on the part of dxFeed.

Please visit www.dxfed.com for latest information updates.

Contents

1.	Introduction	4
1.1.	Purpose	4
1.2.	Scope	4
1.3.	FIX Versions and Messages	4
1.4.	Standard Message Header	4
1.5.	Standard Message Trailer	4
2.	Supported Messages	5
2.1.	Session-Level messages	5
2.1.1.	Logon	5
2.1.2.	Logout	5
2.1.3.	Heartbeat	6
2.1.4.	Test Request	6
2.1.5.	Resend Request	6
2.1.6.	Sequence Reset	7
2.2.	Application-Level messages	7
2.2.1.	Market Data Request	7
2.2.2.	Business Message Reject	10
2.2.3.	Market Data Request Reject	10
2.2.4.	Market Data Incremental Refresh	11
2.2.5.	Market Data Snapshot Full Refresh	12

1. Introduction

1.1. Purpose

This document aims at presenting the details of dxFeed FIX Protocol. The document describes the definition of the various FIX messages.

1.2. Scope

The scope of this document is limited to defining the FIX messages and tags supported by the interface for pricing. It is supposed that the reader already has a comprehensive understanding of the FIX Protocol. This information can be found on the [FIX Protocol website \(www.fixprotocol.org\)](http://www.fixprotocol.org) and is out of the scope of this document.

1.3. FIX Versions and Messages

dxFeed FIX is the financial information exchange protocol. Its messages are based on FIX protocol v4.4.

1.4. Standard Message Header

Tag	Field Name	Required	Comments, valid values
8	BeginString	Y	FIX.4.4
9	BodyLength	Y	Message length excluding the CheckSum field
35	MsgType	Y	Message type
49	SenderCompID	Y	Assigned value used to identify firm sending message
56	TargetCompID	Y	Assigned value used to identify receiving firm
34	MsgSeqNum	Y	Integer message sequence number
43	PossDupFlag	C	Required for retransmitted messages. N = Original transmission, Y= Possible duplicate
52	SendingTime	Y	Time of message transmission (always expressed in UTC) with or without milliseconds depending on FIX session options.
122	OrigSendingTime		Used in message resend

1.5. Standard Message Trailer

Tag	Field Name	Required	Comments, valid values
10	Checksum	Y	Three byte, simple checksum. Always last tag in the message; i.e. serves with the trailing <SOH> as the end-of-message delimiter. Always written as three characters. (e.g. 89 would be '089')

2. Supported Messages

Session protocol assures client identification, sequential request processing, session state control and ability to restore the session after downtime.

In the scope of a session, all FIX messages are identified by unique integer sequence number and processed in that order. When incoming sequence number does not match expected one, the session must be recovered. If incoming sequence number is less than expected and *PossDuplFlag* is not set to Y, it is considered a fatal error, and the connection is dropped by the server. If sequence number of incoming message is greater than next expected number, the Resend Request is issued for missed messages.

dxFeed FIX Protocol has the following specific session-level features (due to transient nature of the market data):

- Session recovery is not supported: the client **must** use the *ResetSeqNumFlag* (141) = Y on each logon to reset sequence. Failure to provide this flag upon logon will result in immediate logout.
- The server always responds with the 'GapFill' message to any Resend Request from the client

2.1. Session-Level messages

2.1.1. Logon

Initiates a connection from client side and approves connection if sent by the server.

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = A
98	EncryptMethod	Y	0 = NONE_OTHER. Encryption should be provided on transport level.
108	HeartBtInterval	Y	Heartbeat interval in seconds
141	ResetSeqNumFlag	Y	'Y' to reset sequence numbers
	StandardTrailer	Y	

2.1.2. Logout

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = 5
58	Text	N	Logout reason
	StandardTrailer	Y	

2.1.3. Heartbeat

Message monitors the status of the connection and identifies if a message was not received. FIX session must send Heartbeat message each HeartBtInterval (108) seconds. If there is no Heartbeat message received after HeartBtInterval (108) + prearranged timeout then the connection should be considered as broken and reconnect should be initiated.

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = 5
112	TestReqID	N	Required when the heartbeat is the result of TestRequest message. Heartbeats issued as the result of TestRequest have to contain the same value as TestReqID in the TestRequest message.
	<Standard Trailer>	Y	

Also, Heartbeat message is used as a reply to Test Request message.

2.1.4. Test Request

Message checks sequence numbers and verifies the connection status. The other endpoint should reply with a Heartbeat, containing the TestReqID (112) from initial TestRequest message.

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = 5
112	TestReqID	N	Identifier included in TestRequest message has to be returned in Heartbeat answer. Any string can be used as the TestReqID (e.g. a timestamp string).
	<Standard Trailer>	Y	

2.1.5. Resend Request

The message recovers inbound session sequence if some message was missed. dxFeed FIX always responds with a *Sequence Reset* message in a Gap Fill mode to client-initiated resent requests.

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = 2
7	BeginSeqNo	Y	Sequence number of the first message in range to be resent
16	EndSeqNo	Y	Sequence number of the last message in range to be resent
	<Standard Trailer>	Y	

2.1.6. Sequence Reset

The message may be used in two modes:

- **Reset Mode** forces counterparty to adjust inbound message sequence, FillGapFlag=N or omitted
- **Fill Gap Mode** is used during retransmission of messages missed by counterparty. Administrative messages and rejected business messages are not to be retransmitted. Instead Sequence Reset message with FillGapFlag=Y is be used. dxFeed FIX always responds with a Gap Fill sequence reset to resend requests.

Tag	Field Name	Required	Comments, valid values
	<Standard Header>	Y	MsgType = 4
123	GapFillFlag	N	N = sequence reset, the counterparty must adjust inbound sequence number. Y = indicates the message is used instead of administrative or business messages which are not to be resent
36	NewSeqNo	Y	Adjusted sequence number
	<Standard Trailer>	Y	

2.2. Application-Level messages

dxFeed FIX system supports the following client-originated messages:

- **In: Market Data Request** – this message is sent by a client to subscribe / unsubscribe / receive a snapshot of market data (quotes and candles).

2.2.1. Market Data Request

Tag	Field Name	Required	Data Type	Comments
	<Standard Header>	Y		MsgType = V
262	MDReqId	Y	String	Unique identifier for Market Data Request assigned by the client system. To unsubscribe, one must send the same ID with tag 263 = 2.
263	SubscriptionRequestType	Y	Char	Data request type. Possible values are: <ul style="list-style-type: none"> • 0 = Snapshot. dxFeed FIX responds with a current snapshot of market data and doesn't send any further data for this request

				<ul style="list-style-type: none"> • 1 = Snapshot + Updates (subscribe). dxFeed FIX-responds with a current snapshot of market data and starts streaming updates • 2 = Unsubscribe. dxFeed FIX-stops sending updates for this request. 	
264	MarketDepth	Y	Int	Depth of market for this request. Possible values are: <ul style="list-style-type: none"> • 0 = full book depth • 1 = top of the book 	
265	MDUpdateType	Y	Int	Specifies the type of market data update. Possible values are: <ul style="list-style-type: none"> • 0 = Full refresh mode • 1 = IncrementalRefresh mode 	
6408	CandleType	N	String	Should only be used for candle subscriptions. Type of the aggregates to request. Possible values are: <ul style="list-style-type: none"> • 1min • 5min • 15min • 30min • 1hour • 2hour • 4hour • Day • Week • Month 	
Component <MDReqGrp>		Y		Repeating group that specifies the type of market data requested.	
267	NoMDEntryTypes	Y	Int	Number of MDEntryType fields requested.	
	269	MDEntryType	Y	Char	Type of data requested. Supported values are: <ul style="list-style-type: none"> • 0 = Bid • 1 = Offer • 4 = Opening price (for candle data) • 5 = Closing price (for candle data) • 7 = High price (for candle data) • 8 = Low price (for candle data)
End Component <MDReqGrp>					

<i>Component</i> <InstrmtMDReqGrp>		Y		Repeating group that specifies instruments for which the market data is requested.	
146	NoRelatedSym	Y	Int	Number of instruments requested	
	55	Symbol	Y	String	Instrument symbol.
<i>End Component</i> <InstrmtMDReqGrp >					
	<Standard Trailer>	Y			

For TopOfBook mode system streams one book entry per side for each instrument. Only the best price for each instrument is delivered, with all prices and amounts aggregated in the best tier.

For FullBookDepth mode system streams set volume tiers (level II quotes), which reflects market depth. Subscription to level II quotes is depended on client setup. If streaming is configured on server side, then nothing to be provided in addition to specifying market depth mode. Otherwise, identifier of pricing stream (MDStreamID) should be provided.

Market data subscriptions are not permanent: client must resubscribe each time a connection is re-established.

Quotes and candles subscription cannot be mixed in one request. A request must contain either set of the fields:

- the Bid / Offer MDEntryTypes and <Instrument> components – in case of subscription for quotes
- the Open / Close / High / Low MDEntryTypes and <Aggregate Specification> components – in case of subscription for aggregated candle data.

A request will be rejected if any of the following is true:

- Request contains both one of (Bid / Offer) and one of the (Open / Close / High / Low) MDEntryTypes
- Request doesn't contain <Aggregate Specification> components but specifies one of the (Open / Close / High / Low) MDEntryTypes
- Request contains CandleType field but specifies one of the (Bid / Offer) MDEntryTypes

dxFeed FIX system supports the following server-originated messages:

- **Out: Business Message Reject** – sent as a response to an application-level message, which fulfills session-level rules and cannot be replied with a normal response.
- **Out: Market Data Request Reject** – sent when a market data request couldn't be honored for technical or business reasons
- **Out: Market Data Incremental Refresh** – delivers market data updates for the client's subscription.
- **Out: Market Data Snapshot** – delivers full snapshot for the market data requested via *MarketDataRequest* message.

2.2.2. Business Message Reject

Client rejects application level messages, which fulfils session-level rules and cannot be rejected via any other means. For example, client sends this message as a reply to any application level message received prior to a Trading Session Status is Open.

Tag	Field Name	Required	Data Type	Comments
	<Standard Header>	Y		MsgType = j
45	RefSeqNum	N	Int	Sequence number of the rejected message
372	RefMsgType	Y	MsgType	Type of the rejected message
380	BusinessRejectReason	Y	Int	Reason for the rejection. The following values are supported: <ul style="list-style-type: none"> • 0 - Other • 1 – Unknown ID • 2 – Unknown Security • 3 – Unsupported Message Type • 5 – Conditionally Required Field Missing
58	Text	N	String	Message to explain reason for rejection; supplied in case 380 = 0.
	<Standard Trailer>	Y		

2.2.3. Market Data Request Reject

Tag	Field Name	Required	Data Type	Comments
	<Standard Header>	Y		MsgType = Y
262	MDReqID	Y	String	Refers to the ID of the request being rejected
281	MDReqRejReason	N	Char	Reason for the rejection. Supported values are: <ul style="list-style-type: none"> • 0 – Unknown Symbol • 1 – Duplicate MDReqID • 2 – Insufficient Bandwidth • 3 – Insufficient Permissions • 4 – Unsupported SubscriptionRequestType • 5 – Unsupported MarketDepth • 6 – Unsupported MDUpdateType • 8 – Unsupported MDEntryType

41	Text	N	String	Free format text describing a reason for rejection
	<Standard Trailer>	Y		

2.2.4. Market Data Incremental Refresh

Tag	Field Name	Required	Data Type	Comments	
	<Standard Header>	Y		MsgType = X	
262	MDReqID	Y	String	Unique identifier for Market Data Request assigned by the client system.	
Component <MDIncGrp>		Y			
268	NoMDEntries	Y	Int	Number of entries in the market data message.	
	279	MDUpdateAction	Y	char	Market data update action. The only supported value for incremental refresh is <ul style="list-style-type: none"> • 0 = New • 1 = Change • 2 = Delete
	269	MDEntryType	Y		Type of data. Supported values are: <ul style="list-style-type: none"> • 0 = Bid • 1 = Offer • 2 = Trade • 4 = Opening price (for candle data) • 5 = Closing price (for candle data) • 7 = High price (for candle data) • 8 = Low price (for candle data)
	278	MDEntryID	Y	String	Unique Market Data Entry identifier.
	1500	MDStreamID	N	String	The identifier or name of the price stream.
	6408	CandleType	N	String	The identifier of candle period only sent for candle updates.
	55	Symbol	Y	String	Instrument symbol.
	270	MDEntryPx	Y	Price	Price of the entry

	271	MDEntrySize	Y	Qty	Quantity or volume represented by the Market Data Entry. Will contain zero for aggregate (candle) data.
	272	MDEntryDate	Y	UTC date	Date of the Market Data Entry
	273	MDEntryTime	Y	UTC time	Time of the Market Data Entry
	451	NetChgPrevDay	N	PriceOffset	If the system is configured to distribute net change data, contains net change value from previous day's closing price vs. this market data update. The value is always rounded to 6 decimal digits.
	6409	NetChgPrevDayPercent	N	PriceOffset	If the system is configured to distribute net change data, contains net change value from previous day's closing price vs. this market data update (as a percent of previous day close price). The value is always rounded to 6 decimal digits.
End Component <MDIncGrp>					
	<Standard Trailer>		Y		

2.2.5. Market Data Snapshot Full Refresh

Tag	Field Name	Required	Data Type	Comments
	<Standard Header>	Y		MsgType = W
262	MDReqID	Y	String	Unique identifier for Market Data Request assigned by the client system.
911	TotNumReports	N	Int	Total number of reports returned in response to a request. Present only in quotes snapshots.
1500	MDStreamID	N	String	The identifier or name of the price stream.
6408	CandleType	N	String	The identifier of candle period, only sent for candle data
55	Symbol	Y	String	Instrument symbol.
451	NetChgPrevDay	N	PriceOffset	If the system is configured to distribute net change data, contains net change value from previous day's closing price

					vs. this market data snapshot. The value is always rounded to 6 decimal digits.
6409	NetChgPrevDayPercent	N	PriceOffset		If the system is configured to distribute net change data, contains net change value from previous day's closing price vs. this market data snapshot (as a percent of previous day close price). The value is always rounded to 6 decimal digits.
Component <MDFullGrp>		Y			Full snapshot of market data for the instrument.
268	NoMDEntries	Y	Int		Number of entries in the market data message.
	269	MDEntryType	Y		Type of data. Supported values are: <ul style="list-style-type: none"> • 0 = Bid • 1 = Offer • 2 = Trade • 4 = Opening price (for candle data) • 5 = Closing price (for candle data) • 7 = High price (for candle data) • 8 = Low price (for candle data) • J = Empty book (when there are no quotes present)
	278	MDEntryID	Y	String	Unique Market Data Entry identifier
	270	MDEntryPx	N	Price	Price of the entry. This field is absent when tag 269 is 'Empty book'.
	271	MDEntrySize	N	Qty	Quantity or volume represented by the Market Data Entry. Will contain zero for aggregate (candle) data. This field is absent when tag 269 is 'Empty book'.
	272	MDEntryDate	Y	UTC date	Date of the Market Data Entry
	273	MDEntryTime	Y	UTC time	Time of the Market Data Entry
End Component <MDFullGrp>					
	<Standard Trailer>	Y			

The API will send the most recent candle as a snapshot.