NFX FIX

REFERENCE DATA

Version 1.7 | **2017-11-04**

CONFIDENTIALITY/DISCLAIMER

1 CONTENTS

2 REFERENCES

[1]

FIX 5.0 SP2 Protocol Specification

http://fixprotocol.org/specifications/fix5.0sp2spec

3 OVERVIEW

This document contains the specification for the FIX interface to NFX Reference Data.

The interface is based on the FIX Protocol 5.0 SP2 standard (Financial Information exchange). More detailed information about the standard can be found in FIX specification document see [1].

The interface implemented by NFX follows the FIX specifications as far as possible. In the majority of cases the structure and semantics of the messages are identical to the standard.

In some cases, the protocol has been extended to cover functions not considered by the standard. These extensions are clearly detailed in the document. In other cases, the standard is ambiguous or indicates that the details should be bilaterally agreed between the parties. In these cases this manual provides a detailed description to avoid any possible ambiguity.

To avoid possible duplication in the sources of information, this document does not include explanations of those matters that comply exactly with the standard. Therefore, the standard documentation should be considered as the main source of information for any matter that is not explicitly covered in this manual.

This specification tries not to repeat what is specified in the FIX standard. In many cases however, the FIX standard is, by necessity, more generic than that required for a specific marketplace. In other cases NFX has found reason to clarify matters. NFX tries to be explicit on deviations from the FIX standard specification in order to avoid confusion.

3.1 SUPPORTED FIX MESSAGES

3.1.1 Administrative messages

Logon
Logout
Sequence Reset
Resend Request
Test Request
Heartbeat
Reject

3.1.2 Inbound Application messages

Application Message Request

3.1.3 Outbound Application messages

Application Message Request Ack
Security Definition
Security Definition Update Report
Market Definition
Trading Session List

3.2 STREAMING REFERENCE DATA OVER FIX

NFX Market Data over FIX is implemented using Application Sequencing. The Reference Data solution is fully compatible with the Market Data solution. It is possible to configure Reference Data and Market Data to be transported over the same FIX session.

4 THE FIX SESSION

The session layer offers limited standard FIX session support. 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 5 for details.

4.1 COMPANY IDS

The Sender- and TargetCompID define the FIX session. A session can only be active (established) between two hosts simultaneously.

- The Company ID of the marketplace is "NFX". This value must be set on:
  - TargetCompID of inbound transactions
  - SenderCompID of outbound transactions
- The Company ID of the client is the participant ID (NX followed by 5 character firm ID).. This id must be set on:
  - SenderCompID of inbound transactions
  - TargetCompID of outbound transactions

4.2 SENDERSUBID

Each inbound business transaction must have the SenderSubID (tag 50) field set to an authenticated user. See the User Authentication section for details on how to authenticate a user. The SenderSubID on incoming transactions will be echoed back in TargetSubID (tag 57) on outbound transactions.

4.3 USER AUTHENTICATION

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

4.3.1 Renew passwords

A new password may be set by setting the NewPassword (925) field along with the current password in Password. The SessionStatus (1409) field of the Logon returned to the client can be checked to see if the new password was accepted.

4.3.2 Expired passwords

If the password has expired when a client tries to log in, the system will respond with a Logout message with SessionStatus set to 8 – Password expired. To gain access, the client must issue a new Logon message with NewPassword set (along with the expired password in Password). If the new password is not valid, the system will respond with another Logout message. SessionStatus will be set to 3 – New session password does not comply with policy. The client will be able to log in again with another new password.

4.4 LOGON

At Logon, clients are identified by:

- CompIDs
- IP Address

The Logon Username (553) and Password (554) fields are used to authenticate the client. When the client is authenticated, the system responds with a Logon message to the client.

4.5 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 30 s.** A heartbeat interval set lower than 10 seconds will result in a Logout response.

4.6 ENCRYPTION

The system does not support encryption.

4.7 DATATYPES AND REQUIRED FIELDS

This specification does not change the datatype on any fields defined in the standard FIX specification. There may be places where this specification restricts the value range of a field further than specified in standard FIX. This will be clearly marked in the spec.

All fields listed in this specification that are marked as required in the standard specification, are required also in this specification. This document specifies additional fields as required. These fields are marked with a 'Q' in the required column of the message listings.

4.8 CHARACTER ENCODING

The FIX gateway uses standard US ASCII encoding.

4.9 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.

4.10 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 5 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.

4.11 THE STANDARD HEADER

All FIX messages contain a Standard Header. The header contains important information such as session identifiers (CompIDs), sequence numbers and message type and length etc.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
8	BeginString	Y	Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. Valid values: FIXT.1.1
9	BodyLength	Y	Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE.
35	MsgType	Y	Defines message type ALWAYS THIRD FIELD IN MESSAGE.
49	SenderCompID	Y	As specified in separate agreement
50	SenderSubID		Required on inbound transactions. Must be set to a valid authenticated user.
56	TargetCompID	Y	As specified in separate agreement
57	TargetSubID		Should not be populated on inbound transactions. Will contain the value of incoming SenderSubID on outbound transactions.
34	MsgSeqNum	Y	Integer message sequence number.
52	SendingTime	Y	Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT")

4.12 THE STANDARD TRAILER

All FIX messages end with a Standard Trailer. The trailer only includes a simple checksum field. The details on how to calculate the checksum can be found in the standard FIX specification.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
10	CheckSum	Y	

4.13 MESSAGE DETAILS

4.13.1 How to interpret the Required (Req'd) column

A 'Y' marks the field as required in standard FIX (and of course also in this implementation). A 'Q' means that the field is required in this implementation although it is not required in standard FIX. No entry at all means the field is optional.

4.13.2 Logon – inbound to Marketplace

The response to a logon is either a Logon, which denotes a successful logon, or a Logout.
A client must be prepared to handle failure scenarios including (but not limited to):

A Logon attempt may fail or be rejected for several reasons. The FIX gateway will react differently depending on the kind of failure. The two different actions it may take are:

Silently ignore the Logon.

- If authentication fails (for security reasons).
- If the wrong Sender or Target CompID is specified.
- For other reasons specified in the standard FIX specifications.
- If the FIX gateway has no connection with the backend system. The gateway relies on the backend to persist its state.

Respond with a Logout.

- Logon failure for other reasons than authentication/security.
- The Logout response to a Logon will always contain a note on why in the Text (58) field.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = A
98	EncryptMethod	Y	Encryption not supported. Valid values: 0 = None / Other
108	HeartBtInt	Y	Heartbeat interval. Any value greater than 10 s is accepted.
141	ResetSeqNumFlag	Q	Indicates both sides of a FIX session should reset sequence numbers. Valid values: Y = Yes **NOTE**: This solution requires resetting the sequence numbers on each Logon. Application Sequencing is used to recover lost messages.
553	Username	Q	User
554	Password	Q	Password.
925	NewPassword		New Password
1137	DefaultApplVerID	Y	The default version of FIX messages used in this session. Valid values: 9 = FIX50SP2
	Standard Trailer	Y	

4.13.3 Logon – outbound from Marketplace

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = A
98	EncryptMethod	Y	Encryption not supported. Valid values: 0 = None / Other
108	HeartBtInt	Y	As specified in inbound Logon. Valid range: Greater than 10 s
141	ResetSeqNumFlag	Q	Indicates both sides of a FIX session should reset sequence numbers. Valid values: Y = Yes **NOTE**: This solution requires resetting the sequence numbers on each Logon. Application Sequencing is used to recover lost messages.
1409	SessionStatus	Q	Status of the FIX session. Valid values: 0 = Session Active 1 = Session password changed
1137	DefaultApplVerID	Y	The default version of FIX messages used in this session. Valid values: 9 = FIX50SP2
	Standard Trailer	Y	

4.13.4 Logout (in/out)

The Logout message is used to gracefully disconnect a FIX session. When receiving a Logout, the counterparty should respond with a Logout. A Logout can also be the response to an unsuccessful Logon attempt.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 5
1409	SessionStatus		Status of the FIX session. This field must only be set by the marketplace. Valid values: 0 = Session logout complete 3 = New session password does not comply with policy 4 = Session logout complete 8 = Password expired 100 = NFX Extension: Invalid body length in received message, session suspended 101 = NFX Extension: Heartbeat interval too low.
58	Text		Free text
	Standard Trailer	Y	

4.13.5 Sequence Reset (in/out)

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 NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 4
123	GapFillFlag		
36	NewSeqNo	Y	
	Standard Trailer	Y	

4.13.6 Resend Request (in/out)

Resend Request is used to recover messages when a sequence number gap has been detected.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 2
7	BeginSeqNo	Y	
16	EndSeqNo	Y	
	Standard Trailer	Y	

4.13.7 Heartbeat (in/out)

A heartbeat message is sent at the interval set at Logon. It is also the response to a Test Request message.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 0
112	TestReqID		Identifier included in Test Request message to be returned in resulting Heartbeat. Required when the heartbeat is the result of a Test Request message.
	Standard Trailer	Y	

4.13.8 Test Request (in/out)

Test Request is used to "ping" the counterparty whenever a heartbeat has not arrived at the negotiated heartbeat interval.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 1
112	TestReqID	Y	Identifier included in Test Request message to be returned in resulting Heartbeat. Required when the heartbeat is the result of a Test Request message.
	Standard Trailer	Y	

4.13.9 Reject (out)

The Reject, or session-level reject, message is sent whenever the FIX gateway is able to at least partially parse the message, but the message does not adhere to the specification and cannot be delivered to the back-end system.

TAG NUM	FIX FIELD NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = 3
45	RefSeqNum	Y	
371	RefTagID		
372	RefMsgType		
373	SessionRejectReason	Q	Valid values: 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 = Comp ID Problem 10 = Sending Time Accuracy Problem 11 = Invalid Msg Type 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 99 = Other
58	Text		
	Standard Trailer	Y	

5 APPLICATION SEQUENCING

5.1 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.

5.1.1 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.

5.1.2 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 5.2.1 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)

5.1.3 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.

5.2 REQUESTING AND RECOVERING REFERENCE DATA

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

5.2.1 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. When recovering a lost connection, the request response will always start at the next sequence number with the real-time data (preceded with a possible snapshot) and disregard the sequence numbers given in the request.

5.2.2 Application Message Request

An Application Message Request is a general request to enable reference data as well as to recover lost reference data. The same message can be used both for recover reference data after a lost connection and to fill out a sequence number gap.

The message contains a repeating group with one entry for each *Application ID (see section 5.1.1 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.

5.2.3 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.

5.2.3.1 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

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

5.2.3.2 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.

6 REFERENCE DATA

6.1 INTRODUCTION

The systems allow the transmission of Security Definitions, Market Definitions 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.

6.2 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 5 for details.

6.3 MAIN WORKFLOW

6.3.1 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.

6.3.2 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).

6.3.3 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.

NFX Extension: 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.

TradSessStatus (340) will contain the type of state.

6.4 MESSAGE DETAILS

6.4.1 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 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
167	SecurityType		Valid values: 1 = Options 3 = Futures 11 = Standard Combination (Strategy)
541	MaturityDate		Date of Maturity.
231	ContractMultiplier		Specifies the ratio or multiply factor to convert from "nominal" units (e.g. contracts) to total units (e.g. shares).
711	NoUnderlyings		Number of underlying instruments. This group is only set if the instrument is derived from an underlying instrument. NOTE: Underlying information will only be set if the underlying is traded within the system.
→	311 UnderlyingSymbol		Underlying identity.
→	309 UnderlyingSecurityID		Orderbook ID of underlying
→	305 UnderlyingSecurityIDSource		Valid values: M = Marketplace-assigned identifier u = Underlying Code (Only used for quote cancels)
→	318 UnderlyingCurrency		Underlying security's Currency.
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 for an Option.
15	Currency		Currency of exercise / subscription / strike price.
762	SecuritySubType		Indicates Strategy subtype. Valid values if defined: COV = Covered Options, for the future leg 637, 1017, 20018 are specified.
526	SecondaryClrID		Minimum Quantity and Quantity Multiple of a TMC.

555	NoLegs				Number of legs (for strategy/combination) instruments. NOTE: Only used for strategies.
20035	LastTradingDate				Date of Last Trading
→	600	LegSymbol			Short name of leg instrument.
→	602	LegSecurityID			Order book ID of leg instrument.
→	603	LegSecurityIDSource			Valid values: M = Marketplace-assigned identifier
→	623	LegRatioQty			The ratio of quantity for this individual leg relative to the entire multileg security.
→	624	LegSide			The side of this individual leg (multileg security). Valid values: B = As Defined C = Opposite
→	637	LegPastPx			The price of the Future leg in TMC for Covered Options.
→	1017	LegOptionRatio			The leg delta of the Future leg in TMC for Covered Options.
→	20018	LegQuantityFuture			Leg Future Quantity Multiple of TMC.
1310	NoMarketSegments			Q	Number of market segments on which the security is traded. Will always be 1.
→	1301	MarketID		Q	Five-character market identifier. Can be mapped to the MarketID in the Market Definition message.
→	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: 1 = Odd Lot 2 = Round Lot 3 = Block Lot A = All or None Lot (NFX Extension enum)
→	→	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.
	Standard Trailer			Y	

6.4.2 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
48	SecurityID		Q	Orderbook ID
22	SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
167	SecurityType			Valid values: 1 = Options 3 = Futures 11 = Standard Combination (Strategy)
541	MaturityDate			Date of Maturity.
231	ContractMultiplier			Specifies the ratio or multiply factor to convert from "nominal" units (e.g. contracts) to total units (e.g. shares).
711	NoUnderlyings			Number of underlying instruments. This group is only set if the instrument is derived from an underlying instrument. **NOTE**: Underlying information will only be set if the underlying is traded within the system.
→	311	UnderlyingSymbol		Underlying identity.
→	318	UnderlyingCurrency		Underlying security's Currency.
→	309	UnderlyingSecurityID		Orderbook ID of underlying
→	305	UnderlyingSecurityIDSource		Valid values: M = Marketplace-assigned identifier u = Underlying Code (Only used for quote cancels)
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 for an Option.
15	Currency			Currency of exercise / subscription / strike price
762	SecuritySubType			Indicates Strategy subtype.

					Valid values if defined: COV = Covered Options, for the future leg 637, 1017, 20018 are specified
526	SecondaryClrID				Minimum Quantity and Quantity Multiple of a TMC
555	NoLegs				Number of legs (for strategy/combination) instruments. **NOTE**: Only used for strategies.
20035	LastTradingDate				Date of Last Trading
→	600	LegSymbol			Short name of leg instrument.
→	602	LegSecurityID			Order book ID of leg instrument.
→	603	LegSecurityIDSource			Valid values: M = Marketplace-assigned identifier
→	623	LegRatioQty			The ratio of quantity for this individual leg relative to the entire multileg security.
→	624	LegSide			The side of this individual leg (multileg security). Valid values: B = As Defined C = Opposite
→	637	LegPastPx			The price of the Future leg in TMC for Covered Options
→	1017	LegOptionRatio			The leg delta of the Future leg in TMC for Covered Options.
→	20018	LegQuantityFuture			Leg Future Quantity Multiple of TMC
1310	NoMarketSegments			Q	Number of market segments on which the security is traded. Will always be 1.
→	1301	MarketID		Q	Five-character market identifier. Can be mapped to the MarketID in the Market Definition message.
→	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: 1 = Odd Lot 2 = Round Lot 3 = Block Lot A = All or None Lot (NFX Extension enum)
→	→	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.
	Standard Trailer			Y	

6.4.3 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	Five-character market identifier.
1396	MarketSegmentDesc	Q	Description of market.
	Standard Trailer	Y	

6.4.4 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
→	340	TradSesStatus	Y	State of Trading Session. Valid values: 1 = Halted 2 = Open 3 = Closed 4 = Pre-Open 101 = Post-Close
→	1237	NoOrdTypeRules		Will always be 0 or 1
→	→	40 OrdType		Shows whether Market orders are allowed in this state. Valid values: 1 = Market
→	1239	NoTimeInForceRules		Will always be 0-2
→	→	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 0 or 1
→	→	1142 MatchAlgorithm		Required in FIX if group is present. Always set to [N/A]
→	→	574 MatchType		Valid values: 4 = Auto-match
→	20032	SessionStateTypeNumber		NFX Extension: The number used in TriggerTradingSessionID on an order to trigger it when this state occurs.
	Standard Trailer		Y	

6.4.5 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	

6.4.6 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 (NFX Extension)
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 (NFX Extension)
58	Text			
	Standard Trailer		Y	

7 APPENDIX A, NFX EXTENSIONS

This chapter details how this solution deviates from standard FIX 5.0 SP2. While great care has been taken to conform to the standard, a number of deviations are unavoidable to support all mechanisms provided by the host.

There are different types of deviations from the standard:

- Fields added. A few user defined fields had to be added to accommodate back-end functionality not present in FIX 5.0 SP2.
- Enumerated values added. Some fields have added enums.
- Field definition changed.

7.1 ADDED FIELDS

FIELD	NAME	ADDED TO MESSAGE	COMMENT
20032	SessionStateTypeNumber	Trading Session List	Data type: int
20035	LastTradingDate	SecurityDefinition, Security Definition Update Report	Date

7.2 ADDED ENUMERATIONS

ENUMERATION	ADDED TO FIELD NAME	COMMENT
100 = Invalid body length in received message, session suspended 101 = Heartbeat interval too low	SessionStatus	
1 = Options 3 = Futures 11 = Standard Combination (Strategy)	SecurityType	
101 = Trade Report Close 102 = System Available	TradSesStatus	
3 = Duplicate requests for application	ApplResponseType	
3 = Duplicate requests for application	ApplResponseError	

7.3 FIELD DEFINITION CHANGED

No field definitions changed.

8 REVISION HISTORY

DATE	REVISION	CHANGE DESCRIPTION
July 3, 2014	1.0	Initial version.
July 3, 2014	1.1	Smaller updates, removing AON
February 5, 2015	1.2	Added that Underlying Order book ID is not applicable for combinations
March 11, 2015	1.3	Minor corrections
April 20, 2015	1.4	Added u as underlying source in Security definition.
May 7, 2015	1.5	Added u in security Definition Update Report.
November 16, 2016	1.6	Copyright and disclaimer change
November 4, 2017	1.7	Copyright and disclaimer change, and added SecuritySubType(762), SecondaryClrID(526), LegLastPx(637), LegOptionRatio(1017), LegQuantityFuture(20018) to Security Definition, Security Definition Update Report.