Ai 1.0 FIX Implementation

EBS Spot Ai

Contents

1 Introduction

The information herein is designated as Confidential Information pursuant to the Customer Agreement in place between EBS and the Customer.

This document outlines the implementation of the FIX 5.0 protocol by the Ai Server version 4.0., which is known as Ai FIX 1.0. As such, it is assumed that the reader has a thorough understanding of the FIX protocol itself, therefore there will be no explanation of the FIX message types or message formats. This information can be acquired through the FIX Protocol website (http://www.fixprotocol.org) and is outside the scope of this document.

The focus is on how the various messages are used by the Ai Server, the information that is included in each message, and the message exchanges between the Ai Client application and the Ai Server. This document shall serve as the proper definition of the Rules of Engagement required to establish Trading Sessions through the Ai Server interface.

The Ai FIX 1.0 implementation will support standard FIX session conventions, however, no recovery data will be provided via the FIX session. All recovery will be handled at the application session level. The life span of the Ai Application Session coincides with that of the FIX Session. Whenever the Ai Application Session is terminated, by whatever means, the FIX Session will also be terminated for that Ai Client. FIX Session sequence numbers are always reset to 1 whenever a new FIX Session is created, which occurs every time the Ai Client connects to the Ai Server using the FIX protocol.

1.1 Ai Server Functionality

The Ai Server functionality can be broken down into several functional areas and messages, as follows:

Logon – Session and Application logon, change password, security list, Session and Application logoff.
Market Data – subscribe, unsubscribe
Orders – order submit, order interrupt, interrupt all, execution report
Deals – trade capture, trade capture request
Session – trade date change, value date change, system up/down, credit low/exhausted, etc

Each of the above mentioned areas can be further broken down into individual message exchanges and message types, as follows:

	Ai Client	FIX Msg	Ai Server	FIX Msg
Logon	Logon Request	BE	Logon Response	BF
	Change Password Request	BE	Change Password Response	BF
	Application Logoff Request	BE	Logoff Response	BF then 5
	FIX Session Logon Request	A	FIX Session Logon Response	A
Market Data	Market Data Subscribe	V	Subscribe Response (snapshot/full refresh)	W
			Subscription Rejection	Y
	N/A		Incremental updates (following the initial subscription)	X
	Market Data Unsubscribe request (no response needed)	V	N/A	
Orders	Submit Order request	D	Execution Report	8
	N/A		Execution report (generated by changes to Order Status)	8
	Order Interrupt	F	Execution Report (with status)	8
			Order Cancel Reject	9
	Interrupt All	q	Execution Report (with status)	8
			Order Cancel Reject	9
Deals	N/A		Trade Capture report (generated by the execution of a trade)	AE
	Trade Capture request (query)	AD	Trade Capture report	AE

	Ai Client	FIX Msg	Ai Server	FIX Msg
			Request Failed	AR
Session	N/A		Session Message (initiated by session events)	h

Note: Each of the FIX message type defined in the above table is described in further detail in the following sections.

2 Header & Footer

The following represent the format of the FIX header and footer, as used by the Ai Server. These fields are the only ones that are used and sent by the Ai Server, and the only ones that the Ai Server expects to see from the Ai Client. Any additional fields that the Ai Client may include in the header will be ignored.

2.1.1 FIX Message Header

The same format is used for both request and response messages.

Tag	Field Name	Reqd	Comments
8	BeginString	Y	Identifies the beginning of the message. Must be the FIRST FIELD IN THE MESSAGE. (Always unencrypted) Valid values: **FIXT.1.1**
9	BodyLength	Y	Message length (number of bytes) forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE. (Always unencrypted)
35	MsgType	Y	Defines message type. ALWAYS THIRD FIELD IN MESSAGE. (Always unencrypted)
49	SenderCompID	Y	Assigned value used to identify sender of message.
56	TargetCompID	Y	Assigned value used to identify receiving firm. Expected value = **ICAP_Ai_Server**
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")

2.1.2 FIX Message Footer

The same format is used for both request and response messages.

Tag	Field Name	Reqd	Comments
10	CheckSum	Y	Three byte, simple checksum (see Volume 2: "Checksum Calculation" for description). ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)

3 FIX Session Logon

3.1 FIX Logon – Request (A)

The Ai Server utilizes the FIX Session Protocol. The FIX Session Logon message type 'A' must be the first message an Ai Client sends to the Ai Server, after establishing a network connection. Ai Server uses this message primarily to solicit and validate the HeartBeat interval. The expected value of HeartBeat interval is 1 second. If different, the Ai Server disconnects the Ai Client citing the Protocol violation.

Ai Server supports version 5.0 of the FIX Protocol, and it is recommended that Ai Clients use the same.

3.1.1 FIX Session Logon Request Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **A** (Session Logon)
98	EncryptMethod	Y	It should always be **0** (None). Ai Application ignores this tag
108	HeartBtInt	Y	It should always be **1**. Ai Application validates this value, and if its different, it sends a protocol violation message, and disconnects the Ai Client
1137	DefaultApplVerID	Y	The default version of FIX being carried over this FIXT session. **7**= FIX5.0. Ai Application doesn't validate this value.
	StandardTrailer	Y	

3.1.2 FIX Session Logon Response Message

If the message is conforming to the FIX Protocol, and has the expected value for the HeartBeat Interval (1 second), Ai Server will simply echo back the Logon message. If the message is non-conforming to the FIX protocol, Ai Server will disconnect the Ai Client.

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **A** (Session Logon)
98	EncryptMethod	Y	It should always be **0** (None). Ai Application ignores this tag
108	HeartBtInt	Y	It should always be **1**. Ai Application validates this value, and if its different, it sends a protocol violation message, and disconnects the Ai Client
1137	DefaultApplVerID	Y	The default version of FIX being carried over this FIXT session. **7**= FIX5.0. Ai Application doesn't validate this value.
	StandardTrailer	Y	

3.1.3 FIX Logon Failed Response Message

The Logon request will fail if the HeartBeat interval is set incorrectly. The Ai Server expects the interval to be set at 1 second. The Ai Server will terminate the FIX session after sending a FIX Logoff message.

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **5**
58	Text	Y	A textual description of the HeartBeat Interval violation.
	StandardTrailer	Y	

4 Application Logon

4.1 Logon Request – User Request (BE)

The **Logon** Request Message is an application level message that is sent by the Ai Client to the Ai Server to log on to the system and begin a new Trading Session. (NOTE: Before the application Logon Request message can be sent, the Ai Server expects a FIX session level l**ogon** to have been performed, as defined and mandated by the Fix Protocol/Engines. The FIX session level logon establishes the connection to the FIX engine and establishes the heartbeat interval for the session, it must be the first message the Ai Client sends.)

4.1.1 Logon Request Message

Tag	Field Name		Reqd	Value
	Standard Header		Y	MsgType = **BE** (Request Message. Initiated by Ai Client)
1129	CstmApplVerID		Y	Ai-FIX version number
923	UserRequestID		Y	Correlation Id created by the Ai Client and used to track other messages associated with this request
924	UserRequestType		Y	**1** = Log On User
553	Username		Y	**username**
554	Password		Y	**password**
5976	NoUserData		N	Number of repeating blocks to follow
→	5977	UserDataName	N	Required if NoUserData is present. Each block will contain one of the following parameter keywords: AutoCancelDuplSession SendConfirmedDeals LargeDifferenceCheck PriceCheck WideSpreadCheck HideMyPrices LocalPriceDisplay
→	5978	UserDataValue	N	Required if NoUserData is present. Each block will contain the value for the parameter specified by the keyword in UserDataName (tag 5977). AutoCancelDuplSession -- boolean SendConfirmedDeals - boolean LargeDifferenceCheck - boolean PriceCheck - boolean WideSpreadCheck - boolean HideMyPrices - boolean LocalPriceDisplay - boolean
	Standard Trailer		Y	

4.1.2 Logon Response

The **Logon** Response message is sent by the Ai Server back to the Ai Client. The **Logon** Response Message includes several session parameters, as shown in the following tables. The response message includes the set of instruments, along with detailed information about the instrument, to which the Ai Client can subscribe and receive market data.

4.1.3 Logon Response Message: Success

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (User Response Message. Sent by Ai Server)
336	TradingSessionID	Y	The Ai session Id from the Ai Server
923	UserRequestID	Y	Carried over from User Request – Logon
553	Username	Y	Carried over from User Request – Logon

Tag		Field Name	Reqd	Comments
926		UserStatus	Y	**1** = Logged In
927		UserStatusText	N	The Logon result status text
5976		**NoUserData**	N	Number of repeating blocks to follow
→	5977	UserDataName	N	Required if NoUserData is present. Each block will contain one of the following parameter keywords: TotalActiveOrders NumberOfOrders NumberOfOrdersTimeInterval AiHostName AiPort
→	5978	UserDataValue	N	Required if NoUserData is present. Each block will contain the value for the parameter specified by the keyword in tag 5977.
146		NoRelatedSym	N	Number of repeating blocks to follow.
→	55	Symbol	N	Required if NoRelatedSym is present. **Base/Local** Denotes the currency pair in CCY1/CCY2 convention.
→	461	CFICode	N	Required if NoRelatedSym is present. **RCSXXX** = FX Spot **FFCNNO** = NDF
→	63	SettlType	N	Required if NoRelatedSym is present. **0** - Regular / FX Spot settlement (T+1 or T+2 depending on currency) **Dx** = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors do not denote business days, but calendar days.
→	561	RoundLot	N	Required if NoRelatedSym is present. Regular trade size for each currency pair
→	75	TradeDate	N	Required if NoRelatedSym is present. Current Trade Date The date will be published in **YYYYMMDD** format
→	9995	SpotValueDateForNDF	N	Conditionally published , may or may not be published when CFICode = FFCNNO (NDF), not published when CFICode=RCSXXX (Fx SPOT) Will contain the SPOT settlement date (also known as the SPOT value date). SPOT settlement date and SPOT value date are equivalent) The date will be published in **YYYYMMDD** format

Tag	Field Name			Reqd	Comments
→	64	SettlDate		N	When CFICode = FFCNNO this field will contain the NDF settlement date When CFICode = RCSXXX this field will contain the SPOT value date The date will be published in **YYYYMMDD** format
→	541	MaturityDate		N	Conditionally published when CFICode = FFCNNO NDF fixing date Not published for SPOT The date will be published in **YYYYMMDD** format
→	9000	NoNestedUserData		N	Required if NoRelatedSym > 0 Number of repeating blocks to follow
→	→	9001	NestedUserDataName	N	Required if NestedNoUsrData is present. Each block will contain one of the following parameter keywords: xPips wideSpread priceIncrement largeDiff minSize maxSize sizeIncrement outsidePriceSpread outsidePriceAmount bestPriceAmount bestPricePlusAmount
→	→	9002	NestedUserDataValue	N	Required if NestedNoUsrData is present. Each block will contain the value for the parameter specified by the keyword in tag 5977
	Standard Trailer			Y	

4.1.4 Logon Failed Response Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (Response Message)
336	TradingSessionID	Y	0 (zero) no trading session was established
923	UserRequestID	Y	Carried over from User Request – Logon
553	Username	Y	Carried over from User Request – Logon
926	UserStatus	Y	The logon result status 2 = Not Logged In
927	UserStatusText	Y	The logon result status text
	Standard Trailer	Y	

4.1.5 Password Expired

If the Ai Client's password has expired, the Ai Client will receive a Password Expired – User Response message from the Ai Server.

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (Response Message)
923	UserRequestID	Y	Carried over from User Request – Logon
553	Username	Y	Carried over from User Request – Logon
926	UserStatus	Y	The logon result status **1000** – Password Expired The Ai Client is not logged on until the password is changed
927	UserStatusText	Y	The logon result status text
	Standard Trailer	Y	

4.1.6 Ai Client's Answer to Password Expired

The Ai Client must respond to a **Password Expired – User** Response message with a Change **Password – User** Request message containing a new password in order to complete the logon sequence.

4.1.7 Change Password Request Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BE** (Request Message)
923	UserRequestID	Y	Ai Client should turn around the UserRequestId received from the initiating Password Expired – User Response message
924	UserRequestType	Y	**3** = Change Password
553	Username	Y	**username**
554	Password	Y	**oldpassword**
925	NewPassword	Y	**newpassword**
	Standard Trailer	Y	

4.1.8 Cancel Duplicate Session User Response Message

The **User** Response message, containing a **UserStatus** of 1001=Cancel Duplicate Session, is sent to the Ai Client in response to a **User Request – Logon** where a spot workstation session already exists for the same user, and the initiating **User Request – Logon** did not specify the **AutoCancelDuplSession** flag, or specifies that flag with a value of false.

Tag	Field Name	Reqd	Value
	Standard Header	Y	MsgType = **BF** (Response Message)
923	UserRequestID	Y	Carried over from User Request – Logon
553	Username	Y	Carried over from User Request – Logon
926	UserStatus	Y	The logon result status **1001** – Cancel Duplicate Session The Ai Client is informed that an existing spot trading workstation session is active for the same user. The Ai Client is required to indicate whether the spot trading workstation session should be canceled in favor of Ai session. The Ai Client does this via the "Answer to Cancel Duplication Session" User Request message, specified below.
927	UserStatusText	Y	The logon result status text
	Standard Trailer	Y	

4.1.9 Ai Client's Answer to Cancel Duplicate Session

The Ai Client must answer the **Cancel Duplicate Session – User** Response message with the **Cancel Duplicate Session – Request** message indicating whether the spot workstation session should be canceled. If the Ai Client indicates that the spot workstation session should not be canceled, by setting the

value of **AutoCancelDuplSession** to false, the Ai Client will be informed that the logon attempt was unsuccessful via the Logon Response (BF) message, and the Ai Client will be disconnected.

Tag	Field Name		Reqd	Value
	Standard Header		Y	MsgType = **BE** (Request Message)
553	Username		Y	Ai Client should turn around the Username received from the initiating Cancel Duplicate Session – User Response message
923	UserRequestId		Y	Ai Client should turn around the UserRequestId received from the initiating Cancel Duplicate Session – User Response message
924	UserRequestType		Y	**1000** = Request to Cancel Duplicate Session
5976	NoUserData		N	Number of repeating blocks to follow
→	5977	UserDataName	N	Required if NoUsrData is present. Each block will contain one of the following parameter keywords: Only the following user data is supported: AutoCancelDuplSession
→	5978	UserDataValue	N	Required if NoUsrData is present. Each block will contain the value for the parameter specified by the keyword in UserDataName (tag 5977): AutoCancelDuplSession -- boolean
	Standard Trailer		Y	

5 Password Change

5.1 Password Change Request

The **Password Change Request** message is a User Request message that is sent by the Ai Client to the Ai Server to initiate a password change.

5.1.1 Password Change Request Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BE** (Request Message)
923	UserRequestId	Y	Correlation Id created by the Ai Client and used to track other messages associated with this request
924	UserRequestType	Y	**3** = Change Password
553	Username	Y	The logon user name
554	Password	Y	The current password
925	NewPassword	Y	The new password
	Standard Trailer	Y	

5.2 Password Change Response

The **Password Change Response** message is sent by the Ai Server in response to a **Password Change Request**.

5.2.1 Password Change Response Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (Response Message)
923	UserRequestId	Y	Carried over from User Request – Password Change Request
553	Username	Y	The logon user name
926	UserStatus	Y	The logon result status **5** – Password Changed
	Standard Trailer	Y	

5.2.2 Password Change Failed Response Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (Response Message)
923	UserRequestId	Y	Carried over from User Request – Password Change Request
553	Username	Y	The logon user name
926	UserStatus	Y	The password changed result status **1002** – Password Change Failed.
927	UserStatusText	N	Result status text indicating the reason for the failure.
	Standard Trailer	Y	

6 Heartbeat

The Ai Server expects the Ai Client application to respond to Heartbeat messages sent by the Ai Server. If the Ai Client fails to respond to Heartbeat messages, and no other message from the Ai Client was received by the Ai Server for a period of 3 seconds, the Ai Server declares the connection **stale**, interrupts all Ai Client's active orders, and unsubscribes the Ai Client from all Market Data. The Ai Server continues to wait for Heartbeats for another 3 seconds after the connection is declared stale (6 seconds total). If the Ai Server receives a Heartbeat from the Ai client during this time, the connection is declared normal, and the Ai Client can resume trading activities (**Note**: Ai Client will need to re-subscribe to Market Views, and re-submit any Orders that were cancelled). If the Ai Client does not respond, the connection is considered dead, and the Ai Server terminates the session and disconnects the Ai Client.

Note: The Ai Server expects the **Heartbeat** interval to have a value of 1 second. In case of a violation, the Ai Server sends a **FIX Logoff** message (as defined by the FIX Protocol), shuts down the physical socket connection, and terminates the FIX Session.

6.1 Heartbeat Message

The format of the heartbeat message as sent by the Ai Server or the Ai Client is identical.

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **0** (Heartbeat)
	Standard Trailer	Y	

6.2 Test Request Message

The Ai Server may send a Test Request message at anytime. The Ai Client must respond to this message with a Heartbeat (MsgType=0) which must include tag 112 with the same value as that received in the Test Request message.

6.2.1 Test Request message format

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **1** (TestRequest)
112	TestReqID	Y	Test Request ID number. Any string can be used as the TestReqID (one suggestion is to use a timestamp string)
	Standard Trailer	Y	

6.2.2 Response to Test Request message

Responding to Test Requests is done using the regular Heartbeat message format (MsgType=0), but tag 112 must be added, and must include the TestReqID value from the Test Request message.

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **0** (Heartbeat)
112	TestReqID	Y	Test Request ID number. The response must use the value of the TestReqID used in the TestRequest message.
	Standard Trailer	Y	

7 Application Logoff

7.1 Application Logoff Request

The **Logoff Request** Message is sent by the Ai Client to the Ai Server to Log out of the system and to terminate the Trading Session. The FIX Session is also terminated as a result of this request.

7.1.1 Application Logoff Request Message

Tag	Field Name	Reqd	Value
	Standard Header	Y	MsgType = **BE** (Request Message. Initiated by Ai Client)
923	UserRequestID	Y	Correlation Id created by the Ai Client and used to track other messages associated with this request
924	UserRequestType	Y	**2** = Logoff User
553	Username	Y	**username**
	Standard Trailer	Y	

7.2 Application Logoff Response

The **Logoff Response** message is sent by the Ai Server back to the Ai Client. It is also initiated by the Ai Server when it detects a Protocol Violation. The **Logoff Response** Message includes the reason for the logoff.

7.2.1 Application Logoff Response Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **BF** (User Response Message. Sent by Ai Server)
923	UserRequestID	Y	Carried over from the User Request – Logoff Request
553	Username	Y	**username**
926	UserStatus	Y	The logoff result status **2** = Not Logged in
927	UserStatusText	Y	The logoff result status text
	Standard Trailer	Y	

The FIX Session is then terminated and the connection is closed.

8 Session Logoff

The Ai Client should always send the "Application Logout Request" message (BE) to logout of the Ai Server. Consequently, Ai Server will send the "Application Logout Response" message followed by the "FIX Session Logout" message, and will then terminate the socket connection.

If an Ai Client sends the "FIX Session Logout" message, while the user is still logged-on, the Ai Server will simply ignore the message.

9 Market Data

9.1 Market Data Subscribe/Unsubscribe

This message is sent by the Ai Client to subscribe to Market Views for specific Currency Pairs. The Ai Server response to a successful **Market Data Request** (V) message is composed of two message types. Each subscribed instrument will result in a single, initial, **Market Data – Snapshot/Full Refresh** (W) message, containing the full set of market data. This will be followed by **Market Data – Incremental Refresh** (X) messages whenever there is a change to the market data. The incremental message can contain updates for multiple instruments in a single message.

The Ai Server must be configured to allow market updates. If disabled, market subscriptions are rejected and the Ai Client application receives no market updates. The Ai Server must also be enabled for either "SPOT", "NDF", or both for subscriptions to be successful.

The Market Data Request message can also be sent by the Ai Client to remove a subscription for a specified instrument list. After this message is processed by the Ai Server, no further market updates will be received by the Ai Client for the designated instruments listed in the message.

Unsubscribing from an instrument that is not already subscribed to will have no effect, and that particular instrument in the unsubscribe list will be ignored. Subscribing to an already subscribed instrument will have no effect; the new subscription attempt for that instrument will simply be ignored.

A subscription message where the NoRelatedSym field's value is < 1 has no effect.

9.1.1 Market Data Subscribe Request Message

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **V** (Subscribe)
262	MDReqID	Y	Correlation Id created by the Ai Client and used to track other messages associated with this request **Note:** For instruments that are successfully subscribed to, the MDReqID in this message will be carried over to the Market Data Snapshot/Full Refresh and Market Data Incremental Refresh messages for those instruments. For instruments that are not successfully subscribed, the MDReqId in this message will be carried over the Market Data Reject message for those instruments. The protocol does not issue a response for currently subscribed instruments that are later unsubscribed
263	SubscriptionRequestType	Y	**1** = Snapshot + Updates (Subscribe)
264	MarketDepth	Y	Always **0** – This field is required by FIX but this field is not used by Ai and will be ignored.
265	MDUpdateType	Y	Always **1** = Incremental Refresh
267	NoMDEntryTypes	Y	Number of repeating blocks to follow Always **1**
→ 269	MDEntryType	Y	Always '*' (without quotes) to indicate all available types of Quotes.
146	NoRelatedSym	Y	Number of repeating blocks to follow. Must be >= 1.
→ 55	Symbol	Y	**Base/Local** = Currency pair in CCY1/CCY2 convention.
→ 461	CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF

Tag	Field Name		Reqd	Comments
→	63	SettlType	Y	**0** = Regular FX Spot settlement (T+1 or T+2 depending on currency) **Dx** = NDF tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
StandardTrailer			Y	

Subscribing to an instrument multiple times will be effectively the same as subscribing once – subsequent subscriptions to the same instrument are ignored.

9.1.2 Market Data Unsubscribe Request Message

Tag	Field Name		Reqd	Comments
	StandardHeader		Y	MsgType = **V** (Subscribe)
262	MDReqID		Y	Unique Id identifying the request. The Ai Client must provide the same **MDReqID** which was contained in the message used to subscribe to this instrument.
263	SubscriptionRequestType		Y	**2** = Disable previous Snapshot + Update Request (Unsubscribe)
264	MarketDepth		Y	Always **0** – This field is required by FIX but this field is not used by Ai and will be ignored.
267	NoMDEntryTypes		Y	Always **1** – Number of repeating blocks to follow
→	269	MDEntryType	Y	Always '*****' (without quotes) to indicate all available types
146	NoRelatedSym		Y	Number of repeating blocks to follow
→	55	Symbol	Y	**Base**/**Local** = Currency pair in CCY1/CCY2 convention
→	461	CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF
→	63	SettlType	Y	Not required if NoRelatedSym = 0 Required if NoRelatedSym > 0. **0** = Regular FX Spot settlement (T+1 or T+2 depending on currency) **Dx** = NDF tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
StandardTrailer			Y	

9.1.3 Market Data Subscribe Response

The Market Data – Snapshot/Full Refresh message is sent once, as the initial response to a successful subscription request, for each instrument subscribed via the Market Data Request (V) message. The Market Data Snapshot/Full Refresh message will contain the complete set of initial bid and offer values (MDEntryTypes).

9.1.4 Market Data– Snapshot/Full Refresh

Tag		Field Name	Reqd	Comments
		StandardHeader	Y	MsgType = **W (**Snapshot/FullRefresh)
262		MDReqID	Y	Will contain the MDReqID carried over from the original Market Data Request subscription message.
55		Symbol	Y	**Base**/**Local** = Currency pair in CCY1/CCY2 convention
461		CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF
63		SettlType	Y	**0** = Regular FX Spot settlement (T+1 or T+2 depending on currency) **Dx** = NDF tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
268		NoMDEntries	Y	Number of repeating blocks to follow. Must be >= 1.
→	269	MDEntryType	Y	Ai FIX will use one for the following enum values from the MDEntryType description. Coupled with MdElementName for offers, bids, and trades (paid, given). **0** = Bid **1** = Offer **2** = Trade (paid, given)
→	5450	MDElementName	Y	Custom field – see enum values table for MDElementName. Specifies the type of update for this entry. The field is defined as a set of enumerated values providing one to one mapping of market data elements to entries in FIX messages.
→	276	QuoteCondition	N	This field is conditionally provided with the message when the following conditions exist which are related to the price contained in the MDEntryPx field. **1000** - No market activity **1001** - No data available **1002** - Not Applicable **1003** - Amount threshold exceeded
→	270	MDEntryPx	Y	This field contains a Price This field is a required field as per the FIX protocol. The field QuoteCondition should be checked first to see if it exists in the message. If QuoteCondition exists in the message then the MDEntryPx field value should be interpreted as follows: When QuoteCondition equals 1000, 1001, or 1002, MDEntryPx will be contain a value of 0.0, however MDEntryPx should be ignored. When QuoteCondition equals 1003, MDEntryPx will contain a valid price.
→	277	TradeCondition	N	This field is conditionally provided with the message when the following conditions exist related to the amount contained in the MDEntrySize field. **1000** - No market activity **1001** – No data available **1002 -** Not applicable
→	271	MDEntrySize	N	This field contains an Amount This field is a required field as per the FIX protocol. The field TradeCondition should be checked first to see if it exists in the message. If TradeCondition exists in the message then the MDEntrySize field value should be interpreted as follows: When TradeCondition equals 1000, 1001, or 1002. MDEntrySize will be set to a value of 0, however MDEntrySize should be ignored.

Tag	Field Name		Reqd	Comments
→	453	NoPartyIDs	N	Number of repeating blocks to follow
→	→ 448	PartyId	N	Required if NoPartyIDs >= 1 The 3 character Trader Id
→	→ 447	PartyIdSource	N	Required if NoPartyIDs >= 1 Always **D** – proprietary/custom code
→	→ 452	PartyRole	N	Required if NoPartyIDs >= 1 Always **12** – executing trader = OurDealerID
→	5457	PriceTimestamp	N	PriceTimestamp is used to timestamp when the price is calculated. Conditionally published when MDElementName is equal to **11**-Paid or **12**-Given,
StandardTrailer			Y	

9.1.5 Market Data – Incremental Refresh

This message will be sent subsequent to the **Market Data – Snapshot/Full Refresh** message to inform the Ai Client of market data updates.

Tag	Field Name		Reqd	Comments
	StandardHeader		Y	MsgType = **X** (Incremental Refresh-Update)
262	MDReqID		Y	Will contain the MDReqID carried over from the original Market Data Request subscription message.
268	NoMDEntries		Y	Number of repeating blocks to follow
→	279	MDUpdateAction	Y	**1** = Change (Always) Must be the first field in this repeating block.
→	269	MDEntryType	Y	Ai FIX will use one for the following enum values from the MDEntryType description. Coupled with MdElementName for offers, bids, and trades (paid, given). **0** = Bid **1** = Offer **2** = Trade (paid, given)
→	5450	MDElementName	Y	Custom field – see enum values table for MDElementName. Specifies the type of update for this entry. The field is defined as a set of enumerated values providing one to one mapping of market data elements to entries in FIX messages.
	55	Symbol	Y	**Base/Local** Denotes the currency pair in CCY1/CCY2 convention.
	461	CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF
	63	SettlType	Y	**0** - Regular / FX Spot settlement (T+1 or T+2 depending on currency) **Dx** = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors do not denote business days, but calendar days.

Tag	Field Name		Reqd	Comments
→	276	QuoteCondition	N	This field is conditionally provided with the message when the following conditions exist which are related to the price normally contained in the MDEntryPx field. **1000** - No market activity **1001** – No data available Both the MDEntryPX field and the QuoteCondition field will be present in the message when the QuoteCondition contains the following value: **1003** - Amount threshold exceeded
→	270	MDEntryPx	N	This field is conditionally provided and will contain a price if QuoteCondition does not equal 1000 or 1001.
→	277	TradeCondition	N	This field is conditionally provided with the message when the following conditions exist which are related to the amount normally contained in the MDEntrySize field. If TradeCondition exists in the message then MDEntrySize will be absent, and vice versa. **1000** - No market activity **1001** – No data available
→	271	MDEntrySize	N	This field is conditionally provided and will contain an amount if TradeCondition does not equal 12 or 13.
→	453	NoPartyIDs	N	Number of repeating blocks to follow Will be included with local bids and local offers only.
→	→	448 PartyId	N	Required if NoPartyIDs >= 1 The 3 character Trader Id
→	→	447 PartyIdSource	N	Required if NoPartyIDs >= 1 Always **D** – proprietary/custom code
→	→	452 PartyRole	N	Required if NoPartyIDs >= 1 Always **12** – executing trader = OurDealerID
→	5457	PriceTimestamp	N	PriceTimestamp is used to timestamp when the price is calculated. Conditionally published when MDElementName is equal to **11**-Paid or **12**-Given,
StandardTrailer			Y	

9.2 Market Data Subscribe Reject

Subscribing to an invalid currency pair, or to a currency pair for which market data cannot be provided – for whatever reason, will result in a **Market Data Request Reject** (Y) message containing text based information identifying the instruments in error from the original **Market Data Request** message.

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **Y** (Subscribe Reject)
262	MDReqID	Y	Ai will populate this field with the unique Id provided by the Ai Client in the original Market Data Request message for this instrument.
281	MDReqRejReason	Y	**0** – Unknown symbol
58	Text	N	The text will include the instrument that was rejected.
	StandardTrailer	Y	

9.3 Custom Market Data Fields (Field Enumerations)

9.3.1 Custom Enum Values for MDElementName

Field Name	Tag	Data Type
MDElementName	5450	String

Value	Type
1	best_bid
2	best_offer
11	paid
12	given
45	dealable_bid
46	dealable_offer
47	local_bid
48	local_offer
49	dealable_regular_bid
50	dealable_regular_offer
51	dealable_outside_bid
52	dealable_outside_offer
53	dealable_plus_bid
54	dealable_plus_offer

9.3.2 Custom Enum Values for QuoteCondition and TradeCondition

Field Name	Tag	Data Type
QuoteCondition	276	String
TradeCondition	277	String

Field Name	Value	Description
QuoteCondition	1000	no market activity
	1001	no data available
	1002	not applicable
	1003	Amount threshold exceeded

Field Name	Value	Description
TradeCondition	1000	no market activity
	1001	no data available
	1002	not applicable

10 Orders

10.1 Order Submit Request

This message allows Ai Clients to submit single-sided orders for a selected currency pair. Ai Clients provide a reference number for each order. The same number must be referenced when the order is interrupted. Customer reference numbers should be unique. Subsequently, the Ai Server will either send an **Order Accepted/Execution Report** or an **Order Rejected/Execution Report** message to the Ai Client.

10.1.1 Order Submit Request Message

Tag		Field Name	Reqd	Comments
		Standard Header	Y	MsgType = **D** (New Order Request - Single)
11		ClOrdID	Y	Unique identifier for Order as assigned by the Ai Client (Customer Reference Number)
55	Combination of these three tags define the instrument as per rules of EBS Trading System	Symbol	Y	**Base**/**Local** = Instrument symbol. e.g., EUR/USD
461		CFICode	Y	**RCSXXX** = SPOT **FFCNNO** = NDF
63		SettlType	Y	**0** = SPOT **Dx, Wx, Mx**, **Yx** = NDF, where: **Dx** = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
54	Combination of these three tags define the type of the order	Side	Y	**1** = Buy (Bid) **2** = Sell (Offer)
40		OrdType	Y	**2** = Limit
59		TimeInForce	Y	**1** = Good 'til Cancelled (Quote) **3** = Immediate or Cancel (Hit)
44		Price	Y	Price per unit of quantity
38		OrderQty	Y	Quantity ordered
60		TransactTime	Y	Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT")
		StandardTrailer	Y	

10.2 Order Submit Response

Order events notify Ai Clients about the orders that were either accepted or rejected by the Ai Server. Messages for an accepted order will contain an EBS generated Order ID used to track a Deal associated with order. For each order submitted by the Ai Client, the Ai Server will initially send either an **Order Accepted Execution Report** or **Order Rejected Execution Report** message back to Ai Client.

10.2.1 Order Submit Response Message

Tag		Field Name	Reqd	Comments
		Standard Header	Y	MsgType = **8** (Execution Report)
11		ClOrdID	Y	Unique identifier for Order as assigned by the Ai Client
39		OrdStatus	Y	**0** = Accepted **2** = Filled **4** = Cancelled **8** = Rejected **C** = Expired

Tag	Field Name		Reqd	Comments
37		OrderID	Y	Unique identifier for Order as assigned by the EBS Trading System
55	Combination of these three tags define the instrument as per rules of EBS Trading System	Symbol	Y	**Base/Local** = Instrument symbol. e.g. EUR/USD
461		CFICode	Y	**RCSXXX** = SPOT **FFCNNO** = NDF
63		SettlType	Y	**0** = SPOT **Dx, Wx, Mx**, **Yx** = NDF, where: Dx = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 Mx = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 Wx = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 Yx = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
54	Combination of these three tags define the type of the order	Side	Y	**1** = Buy (Bid) **2** = Sell (Offer)
40		OrdType	Y	**2** = Limit
59		TimeInForce	Y	**1** = Good 'til Cancelled (Quote) **3** = Immediate or Cancel (Hit)
44		Price	Y	Price per unit of quantity
38		OrderQty	Y	Quantity ordered
150		ExecType	Y	**0** = Accepted **4** = Cancelled **8** = Rejected **C** = Expired **F** = Trade (Partial Fill or Fill)
31		LastPx	N	Last Trade Price. This is conditionally required when ExecType=F and OrdStatus=2. It will have a value of 0.0, and should be ignored by the Ai Client.
32		LastQty	N	Last Trade Quantity. This is conditionally required when ExecType=F and OrdStatus=2. It will have a value of 0, and should be ignored by the Ai Client.
1056		CalculatedCcyLastQty	N	This is conditionally required when ExecType=F and OrdStatus=2. It will have a value of 0.0, and should be ignored by the Ai Client.
17		ExecID	Y	Unique identifier of execution message as assigned by EBS Trading System
151		LeavesQty	Y	Quantity open for further execution.
14		CumQty	Y	Total Quantity Filled
103		OrdRejReason	N	Code to identify reason for order rejection (Reject Message only)
58		Text	N	Free format Text String, used to specify the detailed reason for Order Rejection.
		StandardTrailer	Y	

The Status and Purpose of the **Response** message (**Execution Report**) is defined by the value in field **39** and **150**.

Note: The Ai Server also sends an **Order Done/Execution Report** message to the Ai Client when an active order gets either completely or partially filled or when it expires (BUY or SELL orders) or in the cases where no further action is expected by the Ai Server.

10.3 Order Interrupt Request

An **Order Interrupt Request** Message is sent to cancel individual orders. This message allows Ai Clients to interrupt an active quote (bid or offer). If the order is fulfilled or already cancelled before Ai Server receives this message, the Ai Client will get an **Order Cancel Reject** message. It may have or will also receive an **Order event/Execution Report** with completed Order information, for the impacted order.

10.3.1 Order Interrupt Message

Tag	Field Name		Reqd	Comments
	Standard Header		Y	MsgType = **F** (Order Cancel Request)
11	ClOrdID		Y	Unique identifier for the replacement Order as assigned by the Ai Client. Note that this identifier will be used in the ClOrdID field of the Cancel Reject message of the cancel request is rejected.
41	OrigClOrdID		Y	Original ClOrdID of the order that is being requested to be canceled by this message.
55	Combination of these three tags define the instrument as per rules of EBS Trading System	Symbol	Y	**Base/Local** = Instrument symbol, e.g. EUR/USD FIX Reference: V6P21, V4P40
461		CFICode	Y	RCSXXX = SPOT FFCNNO = NDF
63		SettlType	Y	**0** = SPOT **Dx, Wx, Mx**, **Yx** = NDF, where: Dx = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 Mx = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 Wx = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 Yx = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
54	Combination of these three tags define the type of the order as per rules of EBS Trading System	Side	Y	**1** = Buy (Bid) **2** = Sell (Offer)
40		OrdType	Y	**2** = Limit Order
59		TimeInForce	Y	**1** = Good 'til Cancelled (Quote) **3** = Immediate or Cancel (Hit)
60	TransactTime		Y	Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT")
38	OrderQty		Y	Quantity ordered
37	OrderID		N	Unique identifier for Order as assigned by the EBS Trading System. If provided by Ai Client, the order is interrupted by OrderID else by OrigClOrdID.
	StandardTrailer		*Y*	

10.4 Interrupt All Request

An **InterruptAll/Order Mass Cancel Request** message may be initiated by the Ai Client. On receipt of this message, the Ai Server will attempt to interrupt all the active orders submitted by the Ai Client in the EBS Trading System. Subsequently, the Ai Server will send the Ai Client the **Execution Report** for each of the active orders that were impacted.

10.4.1 Interrupt All Message

Tag	Field Name	Reqd	Comments
	Standard Header	Y	MsgType = **q** (Order Mass Cancel Request - InterruptAll)
11	ClOrdID	Y	Unique identifier for Order as assigned by the Ai Client
60	TransactTime	Y	Time of execution/order creation (expressed in UTC (Universal Time Coordinated, also known as "GMT")
530	MassCancelRequestType	Y	**7** = Cancel all orders
	StandardTrailer	Y	

Note: FIX Protocol has a provision to acknowledge an Order Mass Cancel Request with an **Order Mass Cancel Report** message. However, it also states that each order affected should be individually acknowledged with an **Order Execution Report** or an **Order Cancel Reject** message. Currently EBS Spot Ai doesn't have a specific acknowledgement message for an Interrupt All (**Order Mass Cancel Request**), but it does acknowledge each affected order with an "order cancelled" message.

10.5 Order Interrupt Response

The Ai Server will send an **Order Cancelled** / **Execution Report** to the Ai Client for any active order that was either being interrupted by the Ai Client, or when the Ai Client attempted a logoff while the said order was still active in the market or when the EBS Trading System goes down due to any reason whatsoever.

10.5.1 Order Interrupt Response Message

Tag	Field Name		Reqd	Comments
	Standard Header		Y	MsgType = **8** (Execution Report)
11	ClOrdID		Y	Unique identifier for Order as assigned by the Ai Client
39	OrdStatus		Y	**4** = Canceled
37	OrderID		Y	Unique identifier for Order as assigned by the EBS Trading System
55	Combination of these three tags define the instrument as per rules of EBS Trading System	Symbol	Y	**Base/Local** = Instrument symbol e.g. EUR/USD FIX Reference: V6P21, V4P40
461		CFICode	Y	**RCSXXX** = SPOT **FFCNNO** = NDF
63		SettlType	Y	**0** = SPOT **Dx, Wx, Mx**, **Yx** = NDF, where: Dx = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 Mx = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 Wx = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 Yx = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
54	Combination of these three tags define the type of the order	Side	Y	**1** = Buy (Bid) **2** = Sell (Offer)
40		OrdType	Y	**2** = Limit Order
59		TimeInForce	Y	**1** = Good 'til Cancelled (Quote) **3** = Immediate or Cancel (Hit)
44	Price		Y	Price per unit of quantity
38	OrderQty		Y	Quantity ordered
41	OrigClOrdID		Y	Original ClOrdID of the order that is being cancelled by this message
150	ExecType		Y	**4** = Canceled
17	ExecID		Y	Unique identifier of execution message as assigned by EBS Trading System
151	LeavesQty		Y	Quantity open for further execution.
14	CumQty		Y	Total quantity filled.
58	Text		N	Free format Text String. We can use it to specify the reason for Order Cancellation
	StandardTrailer		Y	

10.5.2 Interrupt Failed Message

An **InterruptFailed** event / **Order Cancel Reject** message is sent to the Ai Client if an **OrderInterrupt** cannot be completed. The Ai Server will return the reason for the failure. This message is only in response to individual orders attempting to be interrupted, not for an InterruptAll. On an InterruptAll, for all the outstanding orders, the Ai Client will receive an Execution Report with the Order status of either “Canceled” or “Filled”.

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **9 (**Order Cancel Rejected)
11	ClOrdID	Y	Unique identifier assigned by the Ai Client in the ClOrdID field of the Order Cancel Request or Order Mass Cancel Request message

Tag	Field Name	Reqd	Comments
41	OrigClOrdID	Y	Original ClOrdID of the order that for which the cancel request is being rejected
37	OrderID	Y	Unique identifier for Order as assigned by the EBS Trading System
39	OrdStatus	Y	Identifies current status of order. **8**=rejected
102	CxlRejReason	N	Code to identify reason for cancel rejection. **99**=Other
434	CxlRejResponseTo	Y	**1** = Order Cancel Request
58	Text	N	Free format Text String. We use it to specify the reason for failure
	StandardTrailer	Y	

11 Deals (Trade Capture)

The Ai Server sends deal transaction details to the Ai Client using **Trade Capture Report** messages. Information provided may vary depending on status and circumstances, meaning that some fields are optional.

Deals are the result of the auto-matching process when the EBS Spot trading system identifies a matching condition between a maker and taker. The deal changes status from the initial match through completion. The Ai Server reports to the Ai Client using **Trade Capture Report** messages for all deal status changes.

The result of the initial matching process (deal in "pending" state) is sent to inform the Ai Client of a potential match. Once the details of the deal are communicated between both parties, the deal is confirmed. The exchange of settlement instructions is the final step in the completion of the transaction. This is reported as a "done" deal to the Ai Client. The "MatchStatus" field is used to expose the state of the deal in the **Trade Capture Report**.

Ai Server provides an option for the customer to receive an event message at the time of confirmation. If the customer chooses to receive the confirmation message, the Ai Client application will receive the "**pending**," "**confirmed**" and "**done**" event messages for each deal transaction. If the customer requests that confirmation messages be excluded, the Ai Server will wait for completion of the deal and only send a **Trade Capture Report** message with a "**done**" MatchStatus after the exchange of the settlement instructions between the parties. This option is controlled by an Ai Client configurable parameter called "SendConfirmedDeals", as described in the "Logon Request" section. The default value for this parameter is "true".

The deal amount given in the "**confirmed**" or "**done**" message can be less than the original "**pending**" amount depending on credit relationships between the parties at the time of the transaction.

11.1 Trade Capture Report

The **Trade Capture Report** message is used to report trades between counterparties.

In the following messages, wherever a Date or a DateTime field is required by FIX protocol, but Ai Server does not have a legitimate value, a value of "19111111" (for Date in YYYYMMDD format), and "19111111-11:11:11" (for DateTime in YYYYMMDD-HH:MM:SS format) is sent (e.g. TradeDate(75) field). Ai Clients should ignore these default values.

11.1.1 Trade Capture Report Message

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **AE**
1003	TradeID	N	OurTicketID
17	ExecID	Y	OriginatorID. Exchanged assigned Execution ID (Trade Identifier) Deal Id in Ai.
573	MatchStatus	Y	The status of this trade with respect to matching or comparison. **0** = Compared, matched or affirmed, or done **1** = Uncompared, unmatched, unaffirmed, or unverified **2** = Advisory or alert, or confirmed, awaiting settlement **Z** = Pending, awaiting conclusion
55	Symbol	Y	**Base/Local** Denotes the currency pair in CCY1/CCY2 convention.
461	CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF Only these values are applicable for Ai.

Tag	Field Name	Reqd	Comments
63	SettlType	Y	**0** = SPOT **Dx, Wx, Mx**, **Yx** = NDF, where: Dx = FX tenor expression for "days", e.g. "D5", where "x" is any integer > 0 Mx = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 Wx = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 Yx = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
32	LastQty	Y	Trade Quantity. Deal Size
31	LastPx	Y	Trade Price.
60	TransactTime	N	CompletedDateTime. Time the transaction represented by this Trade Capture Report occurred. Deal Time in Ai.
75	TradeDate	Y	Used when reporting other than current day trades. Indicates date of trade referenced in this message in **YYYYMMDD** format. Absence of this field indicates current day (expressed in local time at place of trade).
64	SettlDate	N	Takes precedence over SettlType value. For SPOT deals, this represents the "Value Date". For NDF's, it is the actual "Settlement Date".
541	MaturityDate	N	Fixing date (local market date) – only for NDFs
9995	SpotValueDateForNDF	N	Spot Value Date of an NDF Deal– only for NDF's.
552	NoSides	Y	Number of sides For Ai, this is always **1**.
→ 54	Side	Y	**1** = Buy **2** = Sell Deal Type in Ai.
→ 37	OrderID	Y	Order ID in Ai.
→ 11	ClOrdID	Y	Customer Order Reference.
→ 453	NoPartyIDs	N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole. For Ai, the possible values are **0**, **1**, or **2**.
→ → 448	PartyID	N	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs is present.
→ → 447	PartyIDSource	N	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs is present. For Ai, it will always be: **D** = proprietary/custom code
→ → 452	PartyRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs is present. Applicable values for Ai: **17** = Counterparty Deal Code **37** = Counterparty Trader Id
→ 1057	AggressorIndicator	Y	OurDealerType **Y** = aggressor **N** = passive
5462	PrimeDealIndicator	N	Prime Deal Indicator flag **0** = For non-Prime Counterparty **1** = For Prime Counterparty
	StandardTrailer	Y	

11.2 Trade Capture Request (Deal Query)

Deal Status Query is a limited query facility for recovering "in-flight" deal information after a disconnection or a restart. When the Ai Server or the Ai Client aborts an Ai Session, the Ai Server may not have been able to send the status of deals in progress or incomplete matches to the Ai Client before the disconnect occurred.

The **Deal Status Query** feature allows the Ai Client application to retrieve the status and details of deals associated with the active orders in the EBS Spot trading system at the time of the failure. In order to use the **Deal Status Query** facility, the Ai Client application must retain one of the IDs associated with the order or deal.

The Ai FIX protocol provides **Trade Capture Report** request, **Trade Capture Report Request Ack**, and **Trade Capture Report** messages. The **Trade Capture Report** messages contain information for the associated deals based on selected criteria (requested by unique customer Reference ID, EBS order ID or deal ID) within a limited timeframe (2 hours).

For a specific order that resulted in multiple deals, a successful **Trade Capture Report Request** message returns all deal information related to that order as long as one of the deals is within the timeframe allowed. The query returns any "unverified" deal regardless of the time the transaction occurred.

If there is no match to a valid report request, a **report Ack** message returns with the total number of **Trade Reports** set to zero. Valid **Report Requests** returning a total number of trades of zero are not an error and is a normal condition that may be encountered.

Trade Capture Report Requests are limited to retrieving deals up to a specified time limit (2 hours). Attempts to retrieve earlier deals will result in a **Report Ack** message with a **Warning** message indicating the deals were out of the time range. The Ai Server rejects **Trade Capture Report Requests** submitted without qualifiers, with more than one qualifier, with incorrectly formatted references, or with invalid deal or order IDs.

The Ai Server rejects **Trade Capture Report Requests** submitted for an Order Reference ID that is non-unique. The Ai Client application receives a **Trade Capture Report Ack** with a zero in the number of **Trade Reports** field and showing the reason as a non-unique Reference ID.

11.3 Trade Capture Report Request

The **Trade Capture Report Request** can be used to, request one or more **Trade Capture Reports** based upon selection criteria provided on the **Trade Capture Report Request**. At least, one of the fields: OrderID, ClOrdID, or ExecID should be specified on the **Trade Capture Report Request** to define the required Trades.

11.3.1 Trade Capture Report Request Message

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **AD**
568	TradeRequestID	Y	Identifier for the trade request
569	TradeRequestType	Y	Type of Trade Capture Report. 1 - Matched trades matching criteria provided on request (ClOrdID, TradeID, and OrderID.) (AI queries always uses 1)
17	ExecID	N	Deal ID
37	OrderID	N	Order ID
11	ClOrdID	N	Order Reference
	StandardTrailer	Y	

11.4 Trade Capture Report Response

The **Trade Capture Report** is sent in response to the **Trade Capture Report Request**. The **Trade Capture Report** will include the information outlined in the message sent to report the deal with the inclusion of fields TradeRequestID and LastRptRequested.

Tag	Field Name	Reqd	Comments
	StandardHeader	Y	MsgType = **AE**
1003	TradeID	Y	OurTicketID
17	ExecID	Y	Exchanged assigned Execution ID (OriginatorID). Deal Id in Ai.
568	TradeRequestID	N	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request Presence of this field with the contents matching a request indicates this is a response to a request.
912	LastRptRequested	N	Indicates if this is the last report in the response to a Trade Capture Report Request **N** = Not last message **Y** = Last message
573	MatchStatus	Y	The status of this trade with respect to matching or comparison. **0** = Compared, matched or affirmed, or done **2** = Advisory or alert, or confirmed, awaiting settlement **Z** = Pending, awaiting conclusion
55	Symbol	Y	**Base/Local** - Denotes the currency pair in CCY1/CCY2 convention.
461	CFICode	Y	**RCSXXX** = FX Spot **FFCNNO** = NDF
63	SettlType	Y	**0** = Regular FX Spot settlement (T+1 or T+2 depending on currency) Dx, Wx, Mx, Yx = NDF, where: **Dx** = NDF tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days.
32	LastQty	Y	Trade Quantity. (Size)
31	LastPx	Y	Trade Price.
60	TransactTime	Y	Time the transaction represented by this Trade Capture Report occurred. Deal Time in Ai.
75	TradeDate	Y	Used when reporting other than current day trades. Indicates date of trade referenced in this message in **YYYYMMDD** format. Absence of this field indicates current day (expressed in local time at place of trade).
64	SettlDate	Y	Takes precedence over SettlType value. For SPOT deals, this represents the “Value Date”. For NDF’s, it is the actual “Settlement Date”.
541	MaturityDate	N	Fixing date (local market date) – only for NDFs
9995	SpotValueDateForNDF	N	Spot Value Date of an NDF Deal– only for NDF’s.
552	NoSides	Y	Number of sides. For Ai, it will always be 1.
→	54 Side	Y	**1** = Buy **2** = Sell Deal Type
→	37 OrderID	Y	Order ID.
→	11 ClOrdID	Y	Customer Order Reference
→	453 NoPartyIDs	Y	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole. For Ai, the possible values are 1, or 2.
→	→ 448 PartyID	Y	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs is present.
→	→ 447 PartyIDSource	Y	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs is present. Applicable value for Ai: **D** = proprietary/custom code

Tag	Field Name			Reqd	Comments
→	→	452	PartyRole	Y	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs is present. **17** = Counterparty Deal Code **37** = Counterparty Trader Id
→	1057	AggressorIndicator		Y	Boolean to express whether aggressor (taker) side of the trade or not. **Y** = aggressor **N** = passive
5462	PrimeDealIndicator			N	PrimeDealIndicator **0** = For non-Prime Counterparty **1** = For Prime Counterparty
	StandardTrailer			Y	

11.5 Trade Capture Report Ack (Failed Request)

The **Trade Capture Ack** message is used to indicate either that no trades were found that matched the selection criteria specified on the **Trade Capture Report Request**, or that the **Trade Capture Report Request** was invalid due to non-unique order reference criteria.
Note: A **Trade Capture Report Ack** is not required and is not sent if one or more **Trade Capture Reports** will be returned in-band immediately.

11.5.1 Trade Capture Report Ack Message

Tag	FieldName	Reqd	Comments
	StandardHeader	Y	MsgType = **AR**
568	TradeRequestID	Y	Identifier for the trade request
569	TradeRequestType	Y	Type of Trade Capture Report. **1** - Matched trades matching criteria provided on request (ClOrdID, TradeID, and OrderID.) (AI queries are always 1)
55	Symbol	Y	This will always have a value of "[N/A]".
749	TradeRequestResult	Y	Result of Trade Request 4000+ Reserved and available for bi-laterally agreed upon user-defined values. Applicable values for Ai: **0** = Successful (default) **99** = unsupported request: e.g. non-unique order reference ID Set to "0" for successful query request or "99" for unsupported request: e.g. non-unique order reference ID. Explanation provided in text field (58).
750	TradeRequestStatus	Y	Status of Trade Request. **0** = Accepted **2** = Rejected (Set to **0** for successful query request or **2** for non-unique order reference ID)
58	Text	N	Used to specify the reason for a query failure
	StandardTrailer	Y	

12 Session

The Ai Server sends **Session** event messages to notify Ai Clients of non-fatal session conditions, including trading system status changes, credit warnings, Trade Date, and Value Date roll-overs.

12.1 Trading Session Status

The **Trading Session Status** Message provides information on the status of the market. With the move to multiple sessions occurring for a given trading party (morning and evening sessions for instance) there is a need to be able to provide information on what product is trading on what market.

12.1.1 Trading Session Status Message

The **Trading Session Status** can provide an optional repeating group of securities that are available for trading during that session.

Tag	FieldName	Reqd	Comments
	StandardHeader	Y	**h** (lowercase)
336	TradingSessionID	Y	Identifier for Trading Session
340	TradSesStatus	Y	State of the trading session. **0** Unknown **1** Halted **2** Open **3** Closed **4** Pre-Open **5** Pre-Close **6** Request Rejected **1000** = Credit Available **1001** = Credit Low **1002** = Credit Exhausted **1003** = EBS Up **1004** = EBS Down **1005** = EBS Notify **1006** = Trading Day Rollover **1007** = Value Date Rollover **1008** = Trade Date Rollover
75	TradeDate	N	Indicates date of trade referenced in this message in **YYYYMMDD** format. Absence of this field indicates current day (expressed in local time at place of trade). This field is required whenever the TradSesStatus is equal to “Trading Day Rollover” and will indicate the new trading day.
55	Symbol	N	**Base/Local** Denotes the currency pair in CCY1/CCY2 convention. This field is required whenever the TradSesStatus field is equal to “Trade Date Rollover” or “Value Date Rollover”
461	CFICode	N	**RCSXXX** = FX Spot **FFCNNO** = NDF

Tag	FieldName	Reqd	Comments
63	SettlType	N	**0** = Regular FX Spot settlement (T+1 or T+2 depending on currency) **1** = Cash (TOD / T+0) **2** = Next Day (TOM / T+1) **6** = Future **B** = Broken date - for FX expressing non-standard tenor, SettlDate (64) must be specified **C** = FX Spot Next settlement (Spot+1, aka next day) **Dx** = NDF tenor expression for "days", e.g. "D5", where "x" is any integer > 0 **Mx** = FX tenor expression for "months", e.g. "M3", where "x" is any integer > 0 **Wx** = FX tenor expression for "weeks", e.g. "W13", where "x" is any integer > 0 **Yx** = FX tenor expression for "years", e.g. "Y1", where "x" is any integer > 0 Noted that for FX the tenors expressed using Dx, Mx, Wx, and Yx values do not denote business days, but calendar days. This field is required whenever the TradSesStatus field is equal to “Trade Date Rollover” or “Value Date Rollover”
64	SettlDate	N	Specifies the settlement date for NDFs. Specifies the SPOT value date. This field is required whenever the TradSesStatus field is equal to “Trade Date Rollover” or “Value Date Rollover”
541	MaturityDate	N	Fixing date (local market date) – only for NDFs This field is required whenever the TradSesStatus field is equal to “Trade Date Rollover” or “Value Date Rollover” and the CFICode indicates NDF.
9995	SpotValueDateForNDF	N	SPOT - This field is required whenever the TradSesStatus field is equal to “Trade Date Rollover” or “Value Date Rollover” and the CFICode indicates NDF.
58	Text	N	Description of warning message – credit and system state warning messages.
	StandardTrailer	Y	

12.2 Trading System Status Change

If the EBS Spot trading system becomes unavailable, it sends a notification to the Ai Client application. When this condition occurs, the Ai session remains active and connected. The EBS Spot trading system interrupts any outstanding orders. The trading system rejects new orders until it is up and running again. The Ai Server automatically sends the “EBSUp” **Trading Session Status** message to an active session when trading resumes.

If the EBS Spot trading system is unavailable when an Ai Client initiates a new session, the authentication and logon will process normally. The Ai Server sends an “EBSDown” **Trading Session Status** message to the Ai Client application indicating that EBS trading is down. The trading system resumes processing orders when it is back to normal and the Ai Client application receives an “EBSUp” **Trading Session Status** message.

There may be a need for the Ai Client to end the Ai session and restart later depending on the duration of the condition.

12.2.1 Spot Status Change Example: EBS Down

Standard Header	
35=h	Message Type
336=<session ID>	TradingSessionID
340=1004	EBS Down
58=EBS Down	Description
StandardTrailer	

12.3 Credit Warnings

Similarly, in situations of depleted or low credit, the Ai Server sends a **Trading Session Status** message to the Ai Client. The Ai Client should contact their EBS TFA to adjust credit manually during the trading session. If the credit is adjusted during the active Ai session, the Ai Server sends a **Trading Session Status** message to the Ai Client indicating credit is now available. Automatic credit replenishments occur on a daily basis at end of day.

Depending on the credit condition and relationships with other parties, the lack of credit may affect deal transactions and market views.

Note: For Prime Customers, the Ai Server does not send a **Trading Session Status** message for credit warnings. The Ai Client application is notified of order rejection based on insufficient credit.

12.3.1 Credit Warning Example

Standard Header	
35=h	Message Type
336=<session ID>	TradingSessionID
340=1001	Credit Low
58=Credit Low	Description
StandardTrailer	

Note: Only for Ai Bank floors, not PTCs

12.4 End of Day Rollovers

Trading day in the EBS Spot trading system ends at 5:00 pm New York time (1700 hours). This occurs automatically within the system worldwide. At this time, standard floor counterparty credit limits are replenished, internal transaction data capture occurs and the trade date changes to the next business trading date.

Some currencies do not adhere to the Global trade date change (Trade Date Rollover) such as the New Zealand Dollar. If the Ai Client trades in these currencies, the Ai Server can be configured to send individual trade date change messages for each currency in addition to the Global message.

In addition to the trading date, currency pair settlements change at rollover time. Most currency pairs follow the 5:00 pm rollover time, but a few may have a different rollover time.

The Ai client application receives notification and should handle any changes in these values.

12.4.1 Global Trade Date Change Example

When the broker detects a trading day rollover, it sends a message to each connected workstation. The currently active Ai Client application receives a **Trading Session Status** message with the updated trade date.

Standard Header	
35=h	Message Type
336=<session ID>	TradingSessionID
340=1006	Trading Day Rollover
75=<YYYYMMDD>	TradeDate
58=Trading Day Rollover	Description
StandardTrailer	

Note: Occurs only during active session once per trading day. If the Unique End-of-Day parameter is enabled, the Ai Client application will receive a trade date change message for each currency, as well as the Global trade date change.

12.4.2 Individual Trade Date Change Example

Trading Date Rollover for a Currency Pair occurs only during active session, once per trading day per currency pair, only if Non-Standard End-of-Day parameter is enabled

Standard Header	
35=h	Message Type
336=<session ID>	TradingSessionID
340=xx	Trade Date Rollover
75=<YYYYMMDD>	TradeDate
55=<USD/JPY>	Symbol
461=<RSCXXX>	CFICode
63=<C>	SettlType
64=<YYYYMMDD>	SettlDate – Spot value date
541=<YYYYMMDD>	MaturityDate – Fixing date for NDFs only
9995=<YYYYMMDD>	SpotValueDateForNDF – Settlement date for NDFs only
58=Trade Date Change	Description
StandardTrailer	

12.4.3 Value Date Change

Value dates, as described in the “EBS Dealing Rules,” are for transaction settlement dates based on trading business days with recognition of international holidays.

At the start of an Ai trading session, all dealable currency pairs and their current value dates are returned to the Ai Client in the **ApplicationStartup** response message.

When the trading day rolls over at end of day, value dates may change. The Ai Client receives a **Trading Session Status** message for each currency pair with the new value date.

Standard Header	
35=h	Message Type
336=<session ID>	TradingSessionID
340=xx	Value Date Rollover
75=<YYYYMMDD>	TradeDate
55=<USD/JPY>	Symbol
461=<RSCXXX>	CFICode
63=<C>	SettlType
64=<YYYYMMDD>	SettlDate – Spot value date
541=<YYYYMMDD>	MaturityDate – Fixing date for NDFs only
9995=<YYYYMMDD>	SpotValueDateForNDF– SPOT value date for NDFs only
58=Value Date Change	Description
StandardTrailer	

13 Contact Information

Document Version: 1.0
Document Date: September 28, 2007

EBS Dealing Resources, Inc.
Technology & Operations
One Upper Pond Road
Building F, 3rd Floor
Parsippany, New Jersey, USA 07054
Tel. (973) 257-6600

EBS Dealing Resources, Inc. is a Delaware corporation and is a member of the ICAP group of companies.

ICAP Electronic Broking Ltd, ICAP Securities Ltd and ICAP Securities USA LLC – London branch are re-authorized and regulated by the Financial Services Authority. ICAP Electronic Broking Ltd and ICAP Securities Ltd are members of the London Stock Exchange. For further information, please se www.icap.com.

For Customer Support:

Country	Freephone	Direct Dial		Country	Freephone	Direct Dial
Australia	1800121312	+ 44 (0) 20 7029 9344		Malaysia	1800802912	+ 44 (0) 20 7029 9373
Austria	0800291749	+ 44 (0) 20 7029 9345		Mexico	0018666391907	+ 44 (0) 20 7029 9374
Bahrain	80000375	+ 44 (0) 20 7029 9346		Monaco	N/A	+ 44 (0) 20 7029 9375
Barbados	18005340082	+ 44 (0) 20 7029 9347		Netherlands	08000224174	+ 44 (0) 20 7029 9376
Belgium	080010469	+ 44 (0) 20 7029 9348		New Zealand	0800444226	+ 44 (0) 20 7029 9377
Bermuda	18006230166	+ 44 (0) 20 7029 9349		Norway	80011816	+ 44 (0) 20 7029 9378
Brazil	0008110051994	+ 44 (0) 20 7029 9350		Panama	0018886608293	+ 44 (0) 20 7029 9379
Canada	18005762534	+ 44 (0) 20 7029 9351		Peru	080050793	+ 44 (0) 20 7029 9380
Chile	12300200651	+ 44 (0) 20 7029 9353		Philippines	180014410011	+ 44 (0) 20 7029 9381
China	108004400023	+ 44 (0) 20 7029 9354		Poland	008004411343	+ 44 (0) 20 7029 9382
Colombia	01800 9122064	+ 44 (0) 20 7029 9355		Portugal	800844130	+ 44 (0) 20 7029 9383
Cyprus	N/A	+ 44 (0) 20 7029 9356		Russia	74955809410	+ 44 (0) 20 7029 9384
Czech Republic	N/A	+ 44 (0) 20 7029 9357		Singapore	800 852 3666	+ 44 (0) 20 7029 9385
Denmark	80017779	+ 44 (0) 20 7029 9358		South Africa	0800991174	+ 44 (0) 20 7029 9386
Dubai	N/A	+ 44 (0) 20 7029 9359		South Korea	00308440046	+ 44 (0) 20 7029 9387
Finland	0800114424	+ 44 (0) 20 7029 9360		Spain	900974434	+ 44 (0) 20 7029 9388
France	0800908284	+ 44 (0) 20 7029 9361		Sweden	020792749	+ 44 (0) 20 7029 9389
Germany	08001810598	+ 44 (0) 20 7029 9362		Switzerland (D)	0800558443	+ 44 (0) 20 7029 9390
Greece	0080044129654	+ 44 (0) 20 7029 9363		Switzerland (F)	0800551368	+ 44 (0) 20 7029 9391
Hong Kong	800968580	+ 44 (0) 20 7029 9364		Switzerland (I)	0800551369	+ 44 (0) 20 7029 9392
Hungary	0680014347	+ 44 (0) 20 7029 9365		Taiwan	00801 444125	+ 44 (0) 20 7029 9393
Indonesia	001803440095	+ 44 (0) 20 7029 9366		Thailand	18004410152	+ 44 (0) 20 7029 9394
Ireland	1800409190	+ 44 (0) 20 7029 9367		Turkey	80044942788	+ 44 (0) 20 7029 9395
Israel	18009437368	+ 44 (0) 20 7029 9368		United Kingdom	0800446633	+ 44 (0) 20 7029 9396
Italy	800780939	+ 44 (0) 20 7029 9369		Uruguay	0004110052240	+ 44 (0) 20 7029 9397
Japan	0120653638	+ 44 (0) 20 7029 9370		USA	18008723373	+ 44 (0) 20 7029 9398

All calls to Customer Support are recorded.

Email: customersupport@icap.com