ICAP

Ai Lab Guide

EBS Spot Ai©™ version 4.0

Support for Depth of Book, plus Ai XML and Ai FIX Protocols

Contents

1   EBS Ai Lab Overview

To enable EBS FX ®™ Spot Ai customers to test and refine their trading models before implementing a new trading strategy, ICAP developed the EBS FX Lab.

Once a customer codes their trading application to the EBS FX Spot Ai Server application and installs the appropriate networking technology, customers use the EBS FX Lab to test connectivity, timing and price integrity. Customers analyze trade program performance in the lab using EBS FX Spot market data.

1.1   True View of the Market

EBS is the definitive source of reliable, transactional spot FX market data for the professional FX community, derived from the transactional volume of the EBS FX Spot market (daily average over USD 145 billion). Customers receiving EBS FX market updates via EBS FX Spot Ai in the EBS FX Lab can have confidence in the results they generate from testing scenarios.

1.2   Optimal Test Environment

The EBS FX Lab provides a safe and secure test environment to help customers evaluate potential FX Spot market performance for:

- Mathematical models
- Arbitrage models
- Risk management models
- Streaming executable prices

The EBS FX Lab will also provide the ability to test technical aspects of implementation including:

- Connectivity
- Timing
- Price integrity

2   Customer Access Schedule

2.1   Hours of Support

The EBS FX Lab is available on a 24/7 basis. Technical Support is available weekdays from 8 AM to 6 PM EST. Network Support is available weekdays 7 AM to 7 PM EST. Customers will be notified of any downtime due to system maintenance.

For any connectivity issues, the customer should call ICAP Customer Support.

During the product implementation phase, ICAP Global Solutions Delivery is the first line of support for all development issues.

Once trading in the live EBS FX market, the customer should contact ICAP Customer Support for all issues.

The complete list of ICAP Customer Support global contact numbers is on page 19.

2.2   Scheduling

New EBS FX Spot Ai Customers

For new EBS FX Spot Ai customers, access to the lab will be granted on a first-come, first-serve basis. Once space is available, new customers have full access to the lab for a period of six (6) weeks. The EBS FX Lab is not a development environment – therefore customers should complete development before using the lab to test. After approximately 6–8 weeks, customers will be removed from the testing lab server. Contact your Customer Support representative if you find additional lab time is required.

EBS FX Upgrades

When EBS FX upgrades occur to the Ai Server or API, actively trading customers will be granted additional time in the lab to ensure their application is compatible with the upgrades. The length of time allowed for testing will depend on the extent of the upgrade.

ICAP Global Solutions Delivery will contact the customer in advance of all upgrades to discuss testing requirements.

Additional Time

ICAP recognizes that customers may require additional time in the lab to enhance their application or test new models. To accommodate this need, customers may call ICAP Customer Support to schedule time. The customer will be contacted and informed of their test window within 48 hours of the request. Time will be distributed on a first-come, first-serve basis.

2.3   Connectivity Options

ICAP supports and provides both ISDN and VPN access. Both have inherited advantages and disadvantages that the customer should evaluate based on the information provided, preference or accessibility. There are many factors to consider, including cost and equipment required, when making this decision.

Keep in mind that the EBS Network does not introduce latency or derogation of service when connected. Most of the latency or connectivity issues come from the access method used (ISDN facilities or VPN Internet service provider) to get to the Lab. The EBS FX Lab is not utilized for performance measurement,

only application. The more applications customer has running simultaneously the great the risk of having latency problems or connectivity issues.

There is a *Test Network Questionnaire* form that must be accurately completed and approved for either access methods chosen. The ICAP Network Engineering Team uses this information to complete the configurations or implement changes. All requests must be accompanied with a CR or modified original Questionnaire form from GSD.

ISDN

- ICAP configures and ships the Cisco 2600 router to the customer location.

- The customer supplied the connection of Cisco 2600 router in additions to the ISDN facilities.

- The customer is responsible for the ISDN facilities and billing.

- The ISDN connection is a controlled environment point to point, with a minimal amount of security risk.

- Bandwidth restraints and facilities or line impairments can introduce application or connectivity issues.

VPN

- The customer configures and supplies their VPN access equipment and software. (ICAP recommends and supports Cisco.) All other products are the customer's choice and responsibility.

- ICAP does not support NAT.

- ICAP does not guarantee Internet routing infrastructure, security or reliability.

- There are no facilities or line charges.

3   Customer Acceptance Testing

To maintain an orderly market, ICAP asks customers to complete testing for the functions used in their application. To facilitate this process, ICAP provides testing scenarios for connection, market views, order submission, order interrupts and exceptions.

This testing requires real-time communication between the EBS FX Lab and the customer. ICAP Global Solutions Delivery will schedule this test at a mutually convenient time.

The Ai 4.0 release now supports two protocols, the Ai FIX and Ai XML. Support for the Ai XML protocol includes two versions, Ai XML 2.0 and Ai XML 3.0. The Ai Client must identify the protocol they will use in the Login message.

3.1   Scenarios

Following are testing scenarios that to be performed by customers prior to integrating the Ai Server into their production trading environment. The first table includes testing scenarios complement the Ai XML Specification, for both Ai XML 2.0 and Ai XML 3.0. The Ai FIX 1.0 protocol tests are described in further detail in the *EBS Spot Ai 4.0 Developer's Guide*), which describes both the FIX protocol, and the Ai XML protocols (Ai XML 2.0 and Ai XML 3.0) and message formats.

Ai XML Protocol Test Plans

Ai 4.0 Scenario Test Plans for Ai XML ver. 2.0 and ver. 3.0

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
1	**Connecting/Application Startup**			
1.1	Initial Connection	Ask customers to connect to the Ai Test System.	Initiate connection attempt to EBS Ai Test System	Initiate connection attempt to EBS Ai Test System
		Verify customer status on server log.	Verify connection status.	Verify connection status.
1.2	Handlin Heartbeat Message	Start sending heartbeat request messages at 500 millisecond intervals.	Verify that heartbeat messages are coming in.	Verify that heartbeat messages are coming in.
		Verify that the customer is receiving heartbeat messages every 500 milliseconds.		
		Verify that customer is sending valid **heartbeat** responses within 500 milliseconds of original request.	Send **heartbeat** responses.	Send **heartbeat** responses.
		Verify that the customer is still connected.	Maintain heartbeat loop for 2—3 minutes.	Maintain heartbeat loop for 2—3 minutes.
			Maintain heartbeat loop throughout remaining tests.	Maintain heartbeat loop throughout remaining tests.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
1.3	Log In	Ask customer to send Login request.	Send Login request message with ai_protocol_version, userid, and password.	Send Login request message with ai_protocol_version, userid, and password.
		Process Login request message with supplied ai_protocol_version, userid and password.	Wait for **Login** response message.	Wait for **Login** response message.
		Send **Login** response to client with session ID, valid application and session parameters (param list).	Verify AiProtocolVersion, AiHostName and AiPort in **Login** response message.	Verify AiProtocolVersion, AiHostName and AiPort in **Login** response message.
			Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT".	Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT". CCY pair Information.
1.4	Log In with Password Change	Expire customer password.	Select new password to replace existing password and provide to client program.	Select new password to replace existing password and provide to client program.
		Ask customer to send Login request.	Send Login request message with TFA-supplied ai_protocol_version, userid and password.	Send Login request message with TFA-supplied ai_protocol_version, userid and password.
		Process Login request message with supplied ai_protocol_version, userid and password. System should detect expired password.		Remains in "Pending" state until either Login OK or Log Off disconnects.
		Wait for **Login** response message. Check for ChangePassword action field.		
		Send **Login** response with failed status and ChangePassword action.		
		Process **PasswordChange** request. Update password with supplied new password.	Send **PasswordChange** request with old and new password.	Send **PasswordChange** request with old and new password.
		Send **Login** response to client with session ID, valid application and session parameters (param list).	Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT". Verify AiProtocolVersion, AiHostName and AiPort.	Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT". Verify AiProtocolVersion, AiHostName and AiPort.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
1.5	Log In with Duplicate Session	Log in to a Spot workstation with customer's Ai userid.		
		Ask customer to send **Login** request.	Send **Login** request message with TFA-supplied ai_protocol_version, userid and password.	Send **Login** request message with TFA-supplied ai_protocol_version, userid and password.
		Process **Login** request message with supplied ai_protocol_version, userid and password. System will detect duplicate session condition.		
		Send **Login** response with failed status and CancelOtherSession action.	Wait for **Login** response message. Check for CancelOtherSession action field.	Wait for **Login** response message. Check for CancelOtherSession action field.
		Process **CancelDuplSession** request. Cancel the duplicate session.	Send **CancelDuplSession** request with old and new password.	Send **CancelDuplSession** request with old and new password.
		Send **CancelDuplSession** response with status="ok" to client.	Wait for **CancelDuplSession** response message. Verify status="ok".	Wait for **CancelDuplSession** response message. Verify status="ok".
		Send **Login** response to client with session ID, valid application and session parameters (param list).	Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT". Verify AiProtocolVersion, AiHost, and AiPort.	Process **Login** response message. Verify status="ok", Application Type="TRADING" and Market="SPOT". Verify AiProtocolVersion, AiHost, and AiPort.
1.6	Start Application	Ask customer to send the **ApplicationStartup** request. Type should be "TRADING" and Market should be "SPOT".	Send the **ApplicationStartup** request message with required fields.	Send the **ApplicationStartup** request message with required fields.
		Process the request message. Start the requested application.	Wait for the **ApplicationStartup** response message. Verify status="ok".	Wait for the **ApplicationStartup** response message. Verify status="ok".
		Return an **ApplicationStartup** response message with status="ok". Message will contain current "tradeDate", "regSize" and "valueDate" for each currency pair in the list of dealable currency pairs.	Verify the response message includes the list of dealable instruments with "regSize" and "valueDate" for each currency pair as per the message format. Verify today's "tradeDate."	Verify the response message includes the list of dealable instruments with "regSize" and "valueDate" for each currency pair as per the message format. Verify today's "tradeDate."

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
1.7	Log Off	Ask customer to send **Logoff** event message while session is active.	Send **Logoff** event message.	Send **Logoff** event message.
		Process **Logoff** event. Interrupt all active orders. Log off from broker to end session. Disconnect customer connection.	If active orders at time of **Logoff**, customer should verify receipt of order "cancelled" messages.	If active orders at time of **Logoff**, customer should verify receipt of order "cancelled" messages.
			Server sends message to client, "Disconnected by server." Customer is disconnected.	Server sends message to client, "Disconnected by server." Customer is disconnected.
		Ask customer to initiate connection immediately on Logoff.	"Customer will be able to Login".	"Customer will be able to Login".
1.8	Log In with Invalid Ai Protocol Version	Ask customer to send Login request with incorrect protocol version (use "0.0").	Send **Login** request with protocol version "0.0" and valid username and password.	Send **Login** request with protocol version "0.0" and valid username and password.
		Process **Login** request and detect version mismatch. Send error message for protocol violation.	Client application receives error message. Customer is disconnected.	Client application receives error message. Customer is disconnected.
2	**Market Views**			
2.1	Start Subscription	Ask customer to send a **MarketSubscr** request message with one of the dealable currency pairs.	Send a **MarketSubscr** request message with one of the dealable currency pairs retrieved during the application startup.	Send a **MarketSubscr** request message with one of the dealable currency pairs retrieved during the application startup.
		Process **MarketSubscr** request and return **MarketSubscr** response message with status="ok".	Wait for **MarketSubscr** response message. Verify status="ok".	
		Send **Market** event with initial snapshot.	Wait for **Market** event with initial snapshot.	Wait for **Market** event with initial snapshot.
		Start sending **Market** events.		
2.2	Process Market Views	Send **Market** event message(s).	Process **Market** event messages and interpret contents.	Process **Market** event messages and interpret contents. Subsequent market view only includes changed attributes.
			Customer may want to verify changes in market views with either quotes submitted or deals done.	Customer may want to verify changes in market views with either quotes submitted or deals done.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
3	**Depth of Book Market Views**			
3.1	Process Depth of Book – Bid	Enter Bid that qualifies to DealableBestPlusBid for selected CCY pair.	Verify New Market View generated and customer receives with DealableBestPlusBid populated with size up to the masking threshold.	Verify New Market View generated and customer receives with DealableBestPlusBid populated with size up to the masking threshold.
		Enter Bid that qualifies to DealableOutsideBid for selected CCY pair.	Verify New Market View generated and customer receives with DealableOutsideBid populated with size up to the masking threshold.	Verify New Market View generated and customer receives with DealableBestPlusBid populated with size up to the masking threshold.
3.2	Process Depth of Book – Offer	Enter Bid that qualifies to DealableBestPlusOffer for selected CCY pair.	Verify New Market View generated and customer receives with DealableBestPlusOffer populated with size up to the masking threshold.	Verify New Market View generated and customer receives with DealableBestPlusOffer populated with size up to the masking threshold.
		Enter Bid that qualifies to DealableOutsideOffer for selected CCY pair.	Verify New Market View generated and customer receives with DealableOutsideOffer populated with size up to the masking threshold.	Verify New Market View generated and customer receives with DealableBestPlusOffer populated with size up to the masking threshold.
4	**Normal Order Submission and Deal Processing**			
4.1	Submit Buy or Sell Order Resulting in Full Order Complete	Use a Spot workstation to create an active bid/offer to match against customer's test buy/sell.		
		Ask customer to send a buy or sell **OrderSubmit** using the currency pair and price information from the active bid or offer to obtain a match.	Send a buy or sell order submit as per the XML specification. Use currency pair and price information as supplied by EBS.	Send a buy or sell order submit as per the XML specification. Use currency pair and price information as supplied by EBS.
		Validate the submitted order.		
		Send the order acknowledgment in an **Order** event message to the client.	Process the **Order** event message.	Process the **Order** event message.
			Check for status="accepted" in acknowledgment message.	Check for status="accepted" in acknowledgment message.
	a. If option to receive	Deal could match for full amount or partial.	Process the pending deal event message(s).	Process the pending deal event message(s).

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
	"confirmed" Deal event is disabled	Send "pending" **Deal** event message(s) for each matched amount.	Check for status="pending" in **Deal** event.	Check for status="pending" in **Deal** event.
		Send "done" **Deal** event message(s) for each event.	Verify receipt of status="done" for each deal completed.	Verify receipt of status="done" for each deal completed.
		Send "pending" **Deal** event message(s) for each matched amount.	Verify value-date, role, deal-time and Trader-ID fields in the **Deal done** message.	Verify value-date, role, deal-time and Trader-ID fields in the **Deal done** message.
			Process the pending **Deal** event message(s).	Process the pending **Deal** event message(s).
			Check for status="pending" in **Deal** event.	Check for status="pending" in **Deal** event.
			Verify Trader-ID and Role fields in the **Deal** pending message.	Verify Trader-ID and Role fields in the **Deal** pending message.
	b. If option to receive "confirmed" Deal event is enabled	Send "confirmed" **Deal** event message for each deal confirmed.	Check for status="confirmed" in **Deal** event.	Check for status="confirmed" in **Deal** event.
		Send "done" **Deal** event message for each completed deal transaction.	Verify receipt of status="done" for each **Deal** event message.	Verify receipt of status="done" for each **Deal** event message.
	Process Finished Order	Send completed order in an **Order** event with status="done".	Process completed **Order** event.	Process completed **Order** event.
			Verify status="done".	Verify status="done".
			Verify size is less than or equal to total amount done.	Verify size is less than or equal to total amount done.
4.2	Submit Buy or Sell Order Resulting in Partial Hit	Ask customer to send a buy or sell **OrderSubmit** using the currency pair and price information from the active bid or offer (amount=1) to obtain a match for only part of the available quote.	Send a buy or sell **OrderSumit** with order amount=5. Use currency pair and price information as supplied by EBS.	Send a buy or sell **OrderSumit** with order amount=5. Use currency pair and price information as supplied by EBS.
		Validate the submitted order. Send the **Order** "accepted" event message to the client.	Verify the **Order** event message.	Verify the **Order** event message.
		Send "pending" **Deal** event message for matched amount (1 of 5).	Verify "pending" **Deal** event with size=1.	Verify "pending" **Deal** event with size=1.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
		Send "done" **Deal** event message for match completed.	Verify **Deal** event message with status="done" for amount=1, Verify value-date, role, deal-time and Trader-ID fields in the **Deal** "done" message.	Verify **Deal** event message with status="done" for amount=1, Verify value-date, role, deal-time and Trader-ID fields in the **Deal** "done" message.
		Send completed order in an **Order** event message with status="done".	Verify **Order** event message with status="done" for order size=5, done=1, cancelled=4	Verify **Order** event message with status="done" for order size=5, done=1, cancelled=4
4.3	Submit Buy or Sell Order Resulting in Complete Miss	Ask customer to send a buy or sell **OrderSubmit** with no active quote in the market for the currency pair/price.	Send an **OrderSubmit** BUY or SELL for 5 mil. Use currency pair and price information as supplied by EBS.	Send an **OrderSubmit** BUY or SELL for 5 mil. Use currency pair and price information as supplied by EBS.
		Validate the submitted order. Send the **Order** "accepted" event message to the client.	Verify the **Order** event message.	Verify the **Order** event message.
		Verify no match occurs – no deals executed. Buy or Sell order is missed.		
		Send completed order in an **Order** event message with status="done".	Verify **Order** event message with status="done" for order size=5, done=0, cancelled=5	Verify **Order** event message with status="done" for order size=5, done=0, cancelled=5
4.4	Submit Bid or Offer for Automatch	Use a Spot workstation to create an active bid/offer to match against customer's test bid/offer.		
		Ask customer to send a bid or offer **OrderSubmit** event using the currency pair and price information from the active bid to obtain a match.	Send a bid of offer **OrderSubmit** event as per the XML specification. Use currency pair and price information as supplied by EBS.	Send a bid of offer **OrderSubmit** event as per the XML specification. Use currency pair and price information as supplied by EBS.
		Validate the submitted order.		
		Send the order acknowledgment in an **Order** event message to the client.	Process the **Order** event message.	Process the **Order** event message.
			Check for status="accepted" and "intentionalMatch" attribute set to "true" in the **Order** acknowledgment message.	Check for status="accepted" and "intentionalMatch" attribute set to "true" in the **Order** acknowledgment message.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
	a. "Confirmed" Deal Event Message Enabled	Send "pending" **Deal** event message for each automatch event.	Process the pending **Deal** event message(s).	Process the pending **Deal** event message(s).
			Check for status="pending" in **deal** event for each automatch.	Check for status="pending" in **deal** event for each automatch.
		Send "confirmed" **Deal** event message(s) for each automatch event.	Verify Trader-ID and Role fields in the **Deal pending** message.	Verify Trader-ID and Role fields in the **Deal pending** message.
		Send "done" **deal** event message(s) for each automatch event.	Check for status="confirmed" in **deal** event for each automatch.	Check for status="confirmed" in **deal** event for each automatch.
			Verify receipt of status="done" for each automatch.	Verify receipt of status="done" for each automatch.
			Verify Value-Date, Role, Deal-Time and Trader-ID fields in the **Deal Done** message.	Verify Value-Date, Role, Deal-Time and Trader-ID fields in the **Deal Done** message.
5	**Order Interrupts**			
5.1	Submit Below Market Bid	Use a Spot workstation to create an active bid in market.		
		Ask customer to send a bid **OrderSubmit** event using the currency pair with a bid price below market.	Submit below market bid. (Customer should note order reference ID.) Send a bid **OrderSubmit** event as per the XML specification. Use currency pair and price information as supplied by EBS.	Submit below market bid. (Customer should note order reference ID.) Send a bid **OrderSubmit** event as per the XML specification. Use currency pair and price information as supplied by EBS.
5.2	Successful Order Interrupt by Order Reference ID	Validate the submitted order.		
		Send the order acknowledgment in an **Order** event message to the client.	Process the **Order** event message.	Process the **Order** event message.
			Check for status="accepted" and "intentionalMatch" attribute set to "false."	Check for status="accepted" and "intentionalMatch" attribute set to "false."
			Customer bid will remain in the market unmatched.	Customer bid will remain in the market unmatched.
		Ask customer to send **Order interrupt** message.	Send **Order interrupt** message using order reference ID from order submit.	Send **Order interrupt** message using order reference ID from order submit.
		Process **Order interrupt** event.		
		Interrupt the active order.		
		Send **order** event with status="cancelled".	Process **order** event.	Process **order** event.
			Verify status="cancelled".	Verify status="cancelled".

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
			Verify size="active quote amount".	Verify size="active quote amount".
5.3	Intentional Automatch Unsuccessful Order Interrupt	Validate the submitted order.		
		Send the order acknowledgment in an **order** event message to the client.	Process the **order** event message for a Bid/Offer.	Process the **order** event message for a Bid/Offer.
			Check for status="accepted" and "intentionalMatch" attribute set to "false/true."	Check for status="accepted" and "intentionalMatch" attribute set to "false/true."
			Customer order will be completely filled.	Customer order will be completely filled.
			Wait for 5 seconds (the time Ai server keeps order details).	Wait for 5 seconds (the time Ai server keeps order details).
		Ask customer to send **order interrupt** message.	Send **order interrupt** message.	Send **order interrupt** message.
		Process **order interrupt** event.	Process the negative order acknowledgement.	Process the negative order acknowledgement.
		Generate an **interrupt order negative acknowledgement** event message.	Verify the "message type" and the "reason" field in the **interrupt order negative acknowledgement** message. • "Order not active" (if interrupt sent within about five (5) second of the order submission) • "Order not found" (if interrupt sent after approximately five (5) seconds of the order submission) • "Wrong type for interrupt" (if attempting to interrupt a BUY/SELL)	Verify the "message type" and the "reason" field in the **interrupt order negative acknowledgement** message. • "Order not active" (if interrupt sent within about five (5) second of the order submission) • "Order not found" (if interrupt sent after approximately five (5) seconds of the order submission) • "Wrong type for interrupt" (if attempting to interrupt a BUY/SELL)
5.4	Unsuccessful Interrupt	Validate the submitted order.		
		Send the order acknowledgement in an **order** event message to the client.	Process the **order** event message, necessarily Buy/Sell.	Process the **order** event message, necessarily Buy/Sell.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
6	**Deal Status Query Request**			
6.1	Successful Deal Status Requests using a specific deal ID	Process the **deal query** request message.	Send the **deal query** request message, query by deal ID for a deal previously completed.	Send the **deal query** request message, query by deal ID for a deal previously completed.
		Send the **deal query** response message.	Process the **deal query** response message. (**Note**: Deal query result shall contain single deal with the specific deal ID.) Verify deal details.	Process the **deal query** response message. (**Note**: Deal query result shall contain single deal with the specific deal ID.) Verify deal details.
6.2	Deal Status Requests using a specific order ID (returns no data)	Process the **deal query** request message.	Send the **deal query** request message, query by order ID. Use an order ID that was not correct.	Send the **deal query** request message, query by order ID. Use an order ID that was not correct.
		Send the **deal query** response message. Message returns "OK" with an empty set.	Process the **deal query** response message. Response message will contain no deal or order information.	Process the **deal query** response message. Response message will contain no deal or order information.
6.3	Successful Deal Status Requests using a specific order reference ID	Process the **deal query** request message.	Client application executes order with multiple deals resulting. Send the **deal query** request message, query by **OrderRef**. Use a unique Reference ID.	Client application executes order with multiple deals resulting. Send the **deal query** request message, query by **OrderRef**. Use a unique Reference ID.
		Send the **deal query** response message.	Process the **deal query** response message. (**Note:** Deal status query request result shall consist of all deals with matching order reference ID.) Verify query response returns all deals for this order.	Process the **deal query** response message. (**Note:** Deal status query request result shall consist of all deals with matching order reference ID.) Verify query response returns all deals for this order.
6.4	Unsuccessful (not a unique reference ID)	Process **OrderSubmit** message.	Send **OrderSubmit** message request for two orders using same Reference ID.	Send **OrderSubmit** message request for two orders using same Reference ID.
		Process the **Deal query** request message.	Send the **Deal query** request message, query by **OrderRef**.	Send the **Deal query** request message, query by **OrderRef**.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
		Send the **Deal query** response message with status "failed" and description="query for non-unique reference is not supported."	Verify results.	Verify results.
6.5	Exceeded Time Limit		Client will submit **Order requests** for Orders that occurred more than two (2) hours before session.	Client will submit **Order requests** for Orders that occurred more than two (2) hours before session.
		Process the **Deal query** request message.	Send the **Deal query** request message, query by Order Ref.	Send the **Deal query** request message, query by Order Ref.
		Send the **Deal query** response message with status "failed" and description="Requested data exceeds two hour limit."	Verify results.	Verify results.
7	**Password Change**			
7.1	Customer initiated password change. This can be done anytime during a session after the **Login** and before the **Logoff**.	Ask customer to send **PasswordChange** request message with new password after trading begun.	Send **PasswordChange** request message with old and new passwords.	Send **PasswordChange** request message with old and new passwords.
		Process password change request. Send **PasswordChange** response message with status="ok".	Wait for **PasswordChange** response message. Verify status="ok".	Wait for **PasswordChange** response message. Verify status="ok".
8	**Exceptions**			
8.1	**Login** Message out of sequence attributes	Ask customer to enter a **Login** request with user id, password, and Ai protocol version.	Send **Login** request with user id, password, and Ai protocol version.	Send **Login** request with user id, password, and invalid Ai protocol version.
		Send **Logoff** event message with notification to user for protocol violation. **Note:** User message is embedded in warning text.	Client receives message and is disconnected.	Not applicable.
		Disconnect client session.		

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
8.2	Application Startup sent before successful Login Response Received.	Ask customer to enter a **Login** request and immediately send **ApplicationStartup** message.	Send **ApplicationStartup** without any pause or waiting for server message for **Login** successful.	Not applicable.
		Server receives startup before implicit subscriptions process, resulting in protocol violation.	Client receives message and is disconnected.	Client receives message and is disconnected.
		Server sends notification of protocol violation and disconnects client session.		
8.3	Submit malformed XML	Ask customer to connect and log in normally.	Connect, log in and start the application.	Connect, log in and start the application.
		Ask customer to send malformed message.	Send **order** event without currency pair in message.	Send **order** event without currency pair in message.
		Server sends notification of protocol violation and disconnects client session.	Verify disconnect for protocol violation.	Verify disconnect for protocol violation.
8.4	Cancel=0 Generation of this message is based on timing. When an order Interrupt is sent after the deal goes into "pending" state and before the deal goes into "confirmed" or "done" state, the order "Cancel for 0" is generated. **Note:** This message is rare in the EBS FX production environment and may be difficult to reproduce in a test environment.	Submit an order for a huge amount for a currency pair.		
		Ask customer to send multiple orders that match in a single message.	Submit multiple orders that match the EBS FX price in a single message and immediately interrupt the orders.	Submit multiple orders that match the EBS FX price in a single message and immediately interrupt the orders.

	Task	EBS Ai	Ai XML 2.0 Protocol	Ai XML 3.0 Protocol
9	**Session Events – EOD**			
9.1	Trade Date Rollover Notification at 1700 hours NY time (EST/DST).	Send the **Trade Date Rollover** message. The message shall notify clients of change in Trade Date.	Must be logged in to receive the **trade date rollover** notification.	Must be logged in to receive the **trade date rollover** notification.
			Verify appropriate **trade date rollover** message is received at 5:00 PM NYT. Verify new trade date.	Verify appropriate **trade date rollover** message is received at 5:00 PM NYT. Verify new trade date.
9.2	Value Date Rollover Notification	Send the **Value Date Rollover** message. A new session event message is sent to the clients as and when the value date for each currency pair changes.	Must be logged in to receive the **value date rollover** notification.	Must be logged in to receive the **value date rollover** notification.
			Verify appropriate **value date rollover** message are received. **Note:** One message for each currency pair.	Verify appropriate **value date rollover** message are received. **Note:** One message for each currency pair.

4   Information Section

4.1   Default Market View Rates

DealableBestPlusBid

Part of the Depth of Book feature, the **DealableBestPlusBid** gives the price and available amount one pip away from the best dealable price. The size of "one pip" is defined by the EBS trading system for each currency pair.

DealableBestPlusOffer

Part of the Depth of Book feature, the **DealableBestPlusOffer** gives the price and available amount one pip away from the best dealable price. The size of "one pip" is defined by the EBS trading system for each currency pair.

DealableOutsideBid

Part of the Depth of Book feature, the **DealableOutsideBid** indicates the price and cumulative amount available within a defined reasonable trading spread and size. This indicates either the total amount available at the outside price spread or the price at which the outside amount is available, whichever is better.

DealableOutsideOffer

Part of the Depth of Book feature, the **DealableOutsideOffer** indicates the price and cumulative amount available within a defined reasonable trading spread and size. This indicates either the total amount available at the outside price spread or the price at which the outside amount is available, whichever is better.

minSize

The minimum Order size for this Instrument ID (Currency Pair).

plus

Part of the Depth of Book feature, if the depth of the market at that price is greater than the threshold amount that can be reported for that Currency Pair, the plus attribute will be included. For example, if the maximum reportable size for a Currency Pair is 50m, and the actual depth size is 35m, then the size attribute will show 35m. If the actual depth size is 75m, then the size attribute will show 50m, and the **plus** attribute will be included and it will show "true", which will signify that the depth is greater than 50m. Refer to the **Login Response** message for the plus amount for each Instrument ID.

The plus attribute does not appear if the size attribute is not included.

size

Part of the Depth of Book feature, associated with the **DealableBestBid/Offer**, **DealableBestPlusBid/Offer**, and **DealableOusideBid/Offer**, **size** will show the depth of the market at that price up to a certain maximum amount, by Currency Pair. If depth is greater than the maximum amount for that Currency Pair, the parameter will also include the **plus** attribute, as described above.

Not included for **EbsBestBid/Offer**, **DealableRegularBid/Offer**, **LocalBid/Offer**, and **Paid/Given**.

maxSize

The maximum Order size for this Instrument ID (Currency Pair).

sizeIncrement

From the Minimum size Order, size can be increased up to the maximum in increments equal to this amount.

priceIncrement

Order prices must be specified in increments equal to this amount.

outsidePriceAmount

(threshold) The depth of liquidity at the Outside price will be shown up to this amount. If there are 150,000,000 in the market at the Outside price, and the **outsidePriceAmount** is 50,000,000, the **outsidePriceAmount** is shown. If depth of liquidity is less than the **outsidePriceAmount**, then the actual amount is shown.

outsidePriceSpread

(max) The difference in pips between the Best price and the Outside price.

bestPriceAmount

(threshold) The depth of liquidity at the Dealable Best price will be shown up to this amount. If the Dealable Best price is less than this amount, then the actual amount is shown.

bestPricePlusAmount

(threshold) The depth of liquidity at the Dealable Best Price Plus will be shown up to this amount. If the Dealable Best Price Plus is less than this amount, then the actual amount is shown.

Contact Information

Document Version: 4.0a
Document Date: September 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