integral

OCX™

Streaming Orders API

1.0v3

Copyright & Disclaimer

Tuesday, September 22, 2015 1.0v3

3400 Hillview Avenue, Building 4
Palo Alto, CA 94304
United States of America
Tel: +1 (650) 424 4500

Contents

Foreword

About This Document

This document describes the Streaming Orders API that market makers use to stream limit orders into both public and private OCX matching engines.

About OCX

OCX is a new hybrid design FX Exchange. OCX's advanced matching engine combines streaming limit orders, last-look quotes, and resting liquidity taker orders into both public and private matching engines. Market makers are able to distribute all types of liquidity on a specified basis to each matching engine to optimize their competitive offering.

CHAPTER 1

FIX Solution Overview

1 Introduction

This document describes the OCX Streaming Orders API, a FIX-based channel that allows market-making firms to stream orders into OCX matching engines.

This specification is based on the FIX 4.3 standard (http://fixprotocol.org/) that has been adapted to Integral's purposes. Users should have basic knowledge of the FIX protocol and FIX messaging.

> NOTE: OCX supports only the messages, component blocks, and fields that are described in this document. All fields not in this document are ignored by OCX, including fields that are required or conditionally required by the FIX 4.3 protocol.

1.1 Matching Engines

OCX matching engines are located in Equinix's NY4 data center in New York and LD4 data center in London. The matching engines operate independently and generate their own market data and mid-point values.

1.2 Connectivity Details

1.2.1 Integral Data Center Locations

Integral's trading systems are located in the Equinix NY4 and LD4 data centers:

- NY4: 755 Secaucus Road, Secaucus, NJ 07094
- LD4: 2 Buckingham Avenue, Slough, Berkshire, SL1 4NB, United Kingdom

1.2.2 Connectivity Options

- Local cross-connect to Integral's ECN cages in NY4 or LD4 data centers
- Private network connections
- Internet connectivity is over an IPSEC VPN or a pre-negotiated stunnel. Unencrypted connectivity is not available.
- Public Internet connectivity for UAT access and certification only
- All FIX traffic is TCP-based with a unique target (IP port) for each FIX session.
- Multicast traffic does not travel over customer connectivity routes.

1.2.3 Cross Connectivity

- All customer cross-connectivity is 1/10Gpbs Multimode, Single-mode fiber or copper.
- Integral issues a Letter of Authorization (LOA) to connect to Integral's ECN with up to two fiber cross-connects.
- To avoid delays, customers must confirm the correct organization name or third-party agent name to be used in the LOA.
- Send email to noc@integral.com with information including the name of the cross connect customer entity.
- Parallel cross-connects are connected to different access switches for redundancy.
- Cross-connects are statically routed at NY4 while BGP is the preferred routing mechanism at LD4.
- Integral can provide or use a third party RFC-1918 transit network.
- In NY4, Integral prefers that the customer routes via a globally routable unicast IP-address (non-RFC1918), over the transit network. Integral similarly provides a globally routable unicast network over the transit network. Alternatively, an RFC-1918 address space can be agreed on for routing traffic over the transit network.
- In LD4, Integral accepts a customer's registered IP address and BGP ASN. If required, Integral assigns the customer's server-farm IP addresses and BGP ASN.

2 Trading Sessions

2.1 Sessions

The OCX Streaming Orders API operates on a single order session for placing orders and receiving execution reports. Customers initiate the order session to submit, replace, and cancel limit orders.

Messages are transactional and persistent with guaranteed delivery, reflecting their business criticality. The server resends order session messages in response to a resend request from the customer application.

2.2 Session Availability

Order sessions are available from 17:05:00 EST/EDT Sunday through 17:00:00 EST/EDT Friday.

Sessions are off-line for general maintenance from 17:00:00 EST/EDT until 17:05:00 EST/EDT daily Monday through Thursday.

The FIX sequence numbers for all sessions are reset during general maintenance.

To successfully reconnect at 17:05 EST/EDT, customers must send a Logon message with a sequence number of "1" (one). OCX's Logon response also has a sequence number of "1" (one).

OCX sends a Logout message at 17:00 EST/EDT. Upon receipt of this message, customers should disconnect by sending a Logout message. Prior to OCX sending a Logout message, OCX cancels all open orders and sends execution reports to customer applications.

2.3 Intraday Reconnections

For intraday disconnections, Integral recommends using the next sequence number to reconnect to maintain FIX session persistence.

If a customer application requires sequence numbers reset, the application must send a Logon message with a sequence number of "1" (one) and the `ResetSeqNumFlag` (#141) set to "Y". OCX replies with a Logon message with the sequence number set to "1" (one).

3 Supported Message Set

Only the following FIX message types are supported by the OCX Streaming Orders API. Any other message types received by OCX are rejected.

3.1 Supported Session-level Messages

Table 1 Supported Session-level Messages

Direction	Message Name	MsgType (#35)	Message Purpose
In, Out	Logon	A	Customer initiates a FIX session. OCX responds to a successful Logon request.
In, Out	Logout	5	Customer or OCX ends a FIX session.
In, Out	Heartbeat	0	Customer or OCX signals the status of the communication link. Should be sent when no data has been sent for the number of seconds set by `HeartBtInt` (#108).
In, Out	Test Request	1	Customer or OCX queries the status of the connection when no data has been received for twice the number of seconds set by `HeartBtInt` (#108). **NOTE:** The receiving party should always send a Heartbeat message in response to a Test Request message.
In, Out	Session-level Reject	3	Customer or OCX cannot process a message.
In, Out	Resend Request	2	Customer or OCX detects a gap in the number sequence `MsgSeqNum` (#34) and requests retransmission of messages.

3.2 Supported Business Messages

Table 2 Supported Business Messages

Direction	Message Name	MsgType (#35)	Message Purpose
In	New Order - Single	D	Customer submits an order to OCX.
Out	Execution Report	8	OCX confirms receipt of an order, confirms changes to an existing order, relays order status information, and relays fill information for working orders.
In	Order Cancel Request	F	Customer requests the cancelation for all of the remaining quantity of an existing order.
Out	Order Cancel Reject	9	OCX indicates that the Order Cancel Request or Order Cancel/Replace Request cannot be honored.
In	Order Cancel/Replace Request	G	Customer changes the parameters of an existing order.
In	Order Status Request	H	Customer checks the status of an order.
Out	Business Message Reject	j	OCX indicates that the New Order – Single or Order Cancel/Replace Request message has failed business validation. See “Business Rejection Reasons” on page 39.

3.3 Message Validation

OCX validates all FIX messages to ensure the presence of all mandatory and conditional fields and component blocks.

For messages that fail validation, OCX follows the FIX protocol where possible and responds with the following reject messages by message type:

Table 3 Corresponding Reject Message Types

Incoming Message (Rejected)	Outgoing Reject
Logon (#35=A)	Logout (35=5)
New Order - Single (#35=D)	• Execution Report (35=8) • Business Message Reject (35=j)
Order Cancel Request (#35=F)	Order Cancel Reject (35=9)
Order Cancel/Replace Request (#35=G)	• Order Cancel Reject (35=9) • Business Message Reject (35=j)
Order Status Request (#35=H)	Execution Report (35=8)

4 Session Management

4.1 Connect

The OCX Streaming Orders API acts as the acceptor for all FIX sessions. The customer initiates the connection by sending a FIX Logon message. If the connection can be established, OCX responds with a Logon acknowledgement message.

If the Logon fails, OCX responds with a Logout message. The Logout message includes a rejection reason for the failure in the `Text` (#58) field.

4.2 Heartbeat

Both the customer application and OCX send the Heartbeat message to indicate that the connection is active.

The customer application generates a regular heartbeat at the interval defined by the `HeartBtInt` (#108) field in the Logon message or as a response to a Test Request message. Likewise, OCX generates a regular heartbeat at the interval defined by the customer application in the Logon message and also as a response to a Test Request message sent by the customer application.

The Test Request message forces a heartbeat from the receiving system. The receiving system responds to a Test Request message with a Heartbeat message containing the `TestReqID` (#112) sent in the Test Request message.

4.3 Disconnect

Either the customer application or OCX can initiate disconnection by sending a Logout message. The other party should acknowledge this message with a Logout acknowledgement message.

NOTE: OCX cancels all open orders on logout. See "Order Cancelation" on page 14 for details.

5 Orders

The OCX Streaming Orders API supports the following order attributes.

5.1 Order Types

The order type is specified in the `OrdType` (#40) of the New Order - Single (#35=D) message. OCX supports the following order types:

- Limit (#40=2): An order to buy or sell at a specified price or better
- Pegged (#40=P): Only mid-point pegged orders are allowed. The mid-point pegged order is a passive order and floats at the mid-point of the market.

5.2 Time In Force

The order's lifetime is specified with the `TimeInForce` (#59) field on the New Order - Single (#35=D) message. OCX supports the following time-in-force values:

- GTC (#59=1): Good Till Cancel. Remains active until completely filled or is canceled by the customer.
- DAY (#59=0): Day order. Expires at the end of the day at 17:00 EST/EDT.

5.3 Minimum Quantity

The `MinQty` (#110) field is currently not supported by the OCX Streaming Orders API.

5.4 Side

The `Side` (#54) field specifies the side (buy/sell) of the order from the customer's perspective and in terms of the dealt currency (`Currency` (#15)) specified on the order:

- `Side` (#54)=1 (Buy): The customer buys the dealt currency.
- `Side` (#54)=2 (Sell): The customer sells the dealt currency.

5.5 Minimum Size

OCX enforces minimum sizes for orders by currency pair. The list of OCX minimum size requirements is available at the following Web site:
http://ocx.integral.com/trading/instruments

5.6 Icebergs

Customers can decide to show only part of their order to other OCX counterparties with the `MaxShow` (#210) field on the order message. The `MaxShow` (#210) value must be less than the `OrderQty` (#38) value.

5.7 Mid-point Peg Orders

The OCX Streaming Orders API currently supports only mid-point pegged orders. An offset can be specified in pips using the `PegOffsetType` (#836) and `PegOffsetValue` (#211) fields on the order message.

Because each OCX private matching engine consists of a unique liquidity pool, the mid-point of the matching engine's bid/offer may not correspond to the OCX public matching engine's mid-point. Mid-point pegged orders placed with the OCX Streaming Orders API are pegged at the OCX public matching engine's mid-point.

`PegOffsetValue` (#211) can be used to specify a signed amount that is added to the OCX public matching engine's mid-point.

5.8 Client Order ID

The `ClOrdID` (#11) field is a unique identifier assigned by the customer application to every order submitted to OCX. The `ClOrdID` (#11) field must adhere to the following rules:

NOTE: If the `ClOrdID` (#11) field does not conform, OCX may reject the New Order - Single.

- All open orders must have a unique `ClOrdID`.
- An order's `ClOrdID` must remain unique until the order reaches a terminal state (the order is filled, canceled, replaced, or expired).
- After an order reaches a terminal state, its `ClOrdID` can be reused during the next trading day.
- The `ClOrdID` can contain a maximum of 11 characters.
- The `ClOrdID` can contain only numbers and upper-case letters. Symbols, lower-case letters, spaces, tabs, control characters, and special characters are not allowed.

IMPORTANT: The customer application must ensure the uniqueness of the `ClOrdID` (#11) field. OCX does not validate the `ClOrdID` field for uniqueness apart from ensuring that no two active orders can have the same `ClOrdID`. Non-unique `ClOrdID` values for orders can cause issues with downstream systems and with order status reports.

5.9 Price Conventions

OCX follows standard OTC spot market conventions for order price precision to one-tenth of a pip. You can review OCX price conventions at the following Web site:

http://ocx.integral.com/trading/instruments

For orders that exceed these price-precision standards, OCX rounds appropriately (down for buy orders and up for sell orders).

5.10 Price Bands

In order to ensure market integrity, OCX has a price-banding algorithm that subjects all orders to price validation and rejects orders outside the given band. A proprietary algorithm computes the new high and low price limits at the start of every trading session (17:05:00 EST/EDT daily Monday through Thursday).

Based on market conditions and trading activity, OCX may relax or suspend price-band restrictions.

5.11 Term Currency Orders

OCX Streaming Orders API does not currently support term currency orders.

6 Order Management

6.1 Order States

OCX sends Execution Report messages to notify the customer on updates to their orders. The various order states can be interpreted using two fields in the Execution Report:

- `ExecType` (#150): identifies the purpose of the Execution Report.
- `OrdStatus` (#39): conveys the current state of the order.

A typical order in OCX transitions through several different states during its lifecycle.

Table 4 Order States

Order State	ExecType (#150)	OrdStatus (#39)	Description
New	0 = New	0 = New	The order has been successfully processed, entered in the order book, and acknowledged by the matching engine.
Partially Filled	F = Trade	1 = Partially Filled	The order has been filled for part of the amount on the order. The order is still present in the order book with the remaining quantity.
Filled	F = Trade	2 = Filled	The full amount on the order has been filled. The order is no longer in the order book.
Canceled	4 = Canceled	4 = Canceled	The matching engine has successfully processed the cancel request. The order is no longer in the order book.
Expired	C = Expired	C = Expired	The order has reached a terminal state and is no longer in the order book.
Replaced	5 = Replace	5 = Replaced	The matching engine has successfully processed the replace request. The existing order has been canceled and a new order was created and is in the order book.
Order Rejected	8 = Rejected	8 = Rejected	The order was rejected.
Undefined order	I = Order Status	U = Undefined	The order for which status has been requested is not present in the order book.

6.2 Order Cancelation

Order cancelation can occur in the scenarios outlined below for an order that was previously accepted by OCX:

1. Solicited order cancel

The cancel request is initiated by the customer application and the order can be successfully canceled (the order has not been filled).

2. Unsolicited order cancel
   - OCX cancels orders when they expire.
   - OCX cancels all orders when the customer application disconnects from OCX for any reason, intraday or end of day.

6.3 Order Modification

Customers can use the Order Cancel/Replace Request message to modify the following order parameters:

- Quantity (`OrderQty` (#38))
- Price (`Price` (#44))
- Side (`Side` #54))
- Maximum amount to be shown (`MaxShow` (#210))
- Peg parameters (`PegPriceType` (#1094), `PegOffsetType` (#836), `PegOffsetValue` (#211))

6.4 In-Flight Risk Mitigation

In-flight risk mitigation protects from the risk of a resting order being filled between the time the customer application initiates the cancel/replace request and the time the cancel/replace request is actually completed by OCX.

In-flight mitigation is specified by the `IFRMEnabled` (#9501) field on the Order Cancel/Replace Request message.

> NOTE: You must specify in-flight risk mitigation in the cancel/replace request. It is NOT enabled by default.

The table below describes how OCX applies in-flight risk mitigation for cancel/replace requests.

Table 5 In-Flight Risk Mitigation

In-flight risk mitigation enabled?	Condition	Action taken by OCX
Yes `IFRMEnabled` (#9501)=Y	Quantity in order replace message greater than filled quantity	Order quantity in the updated order is set to the new quantity less the filled quantity.
	Quantity in order replace message less than filled quantity	Order quantity in the updated order is set to 0 (order is effectively canceled).
No `IFRMEnabled` (#9501)=N	Quantity in order replace message greater than filled quantity	Order quantity in the updated order is set to new quantity.
	Quantity in order replace message less than filled quantity	Order quantity in the updated order is set to new quantity.

7 Messaging Controls

OCX enforces thresholds on messages and other administrative operations to:

- Support only valid trading activity.
- Prevent a malfunctioning trading system from impacting the network.

7.1 Messaging Thresholds

Messaging thresholds are specified as messages per second over a pre-defined interval.

The thresholds are in place for all application messages supported by OCX. If a customer application exceeds any of the thresholds, OCX rejects subsequent messages via a Session Level Reject message (#35=3) until the message rate falls below the threshold.

The messaging thresholds are:

Table 6 Messaging Thresholds

Messages	Threshold (messages/second)	Time Interval
Application messages per currency pair per FIX session	100	3 second window
Administrative messages per FIX session	200	3 second window

7.2 Other Administrative Thresholds

OCX automatically closes the ports for any customer session that:

- Exceeds 6 invalid Logon messages (#35=A) in 60 seconds.
- Sends any message other than a Logon message before the session has successfully logged in.

8 Message Formats

This section describes the features common to each message format in the OCX Streaming Orders API.

8.1 Terminology

Each OCX Streaming Orders API message is comprised of message fields defined in this document. Each field has its own set of attributes including:

- Tag: field number
- Field Name: human-readable name for reference
- Req'd: indicates if the field must be sent on the FIX message:
  - Y = required by the FIX 4.3 protocol
  - Y(OCX) = required by the OCX Streaming Orders API implementation
  - N = optional for both OCX and the FIX 4.3 protocol

- C = conditional. Refer to the field's description to determine applicability.
- Valid Values: specific values accepted for this field on the message type
- Description: brief description of the field
- Type(Length): FIX data type of the field and the maximum length of the field value

8.2 Standard Header

Each message type sent over the OCX Streaming Orders API has a standard header component as described below. The header is abbreviated as <Standard Header> in this document.

Table 7 Standard Header Format

Tag	Field Name	Req'd	Valid Values	Description	Type
8	BeginString	Y	FIX4.3	Identifies beginning of new message and protocol version. Must be the first field in the message.	String(7)
9	BodyLength	Y	—	Message length (in bytes) forward to the `CheckSum` (#10) field. Must be the second field in the message.	int(6)
35	MsgType	Y	—	Defines the type of the message. Must be the third field in the message.	String(2)
49	SenderCompID	Y	—	This value is used to identify the firm sending the message. The customer application should set this field with the `SenderCompID` (#49) supplied by OCX for this FIX connection.	String(20)
56	TargetCompID	Y	—	This value is used to identify the firm receiving the message. The customer application should set this field with the `TargetCompID` (#56) supplied by OCX for this FIX connection.	String(20)
34	MsgSeqNum	Y	—	Integer message sequence number.	int(9)
43	PossDupFlag	N	Y=Possible Duplicate N=Original Transmission	Indicates possible retransmission of message with this sequence number. Must be set to "Y" for messages sent in response to a Resend Request.	Boolean(1)

Tag	Field Name	Req'd	Valid Values	Description	Type
52	SendingTime	Y	—	Time the message was transmitted. Always expressed in UTC.	UTCTimestamp(21)

8.3 Standard Trailer

Each message type sent over the OCX Streaming Orders API also has a standard trailer component as described below. The header is abbreviated as <Standard Trailer> in this document.

Table 8 Standard Trailer Format

Tag	Field Name	Req'd	Valid Values	Description	Type
10	Checksum	Y	—	Three byte checksum. Always the last tag in the message. Functions as end-of-message delimiter.	String(3)

9 Administrative Message Formats

9.1 Logon

The Logon message is sent by the customer application to establish a FIX session with OCX. If a Logon message is sent with invalid fields, OCX responds with a Logout message. The `Text` (#58) field of the Logout message contains the reason for the rejection of the Logon message.

9.1.1 Valid Responses

- Logon (#35=A)
- Logout (#35=5)

9.1.2 Message Format

Table 9 Logon Format

Tag	Field Name	Req'd	Valid Values	Description	Type
<Standard Header>		Y	—	35=A	—
98	EncryptMethod	Y	0=None	Always unencrypted	int(1)
108	HeartBtInt	Y	—	Heartbeat interval in seconds. The customer application determines the heartbeat interval.	int(3)
141	ResetSeqNumFlag	N	Y=Yes, reset sequence numbers N=No	Indicates that the both sides of the FIX session should reset sequence numbers.	Boolean(1)

Tag	Field Name	Req'd	Valid Values	Description	Type
553	Username	Y(OCX)	—	This field should be set with the username supplied by OCX for this FIX connection.	String(20)
554	Password	Y(OCX)	—	This field should be set with the password supplied by OCX for this FIX connection.	String(20)
<Standard Trailer>		Y	—	—	—

9.2 Logout

The customer application sends the Logout message to OCX to end a session. OCX sends a single Logout message in response when all Logout related actions have been completed by OCX.

9.2.1 Valid Responses

- Logout (#35=5)

9.2.2 Message Format

Table 10 Logout Format

Tag	Field Name	Req'd	Valid Values	Description	Type
<Standard Header>		Y	—	35=5	—
58	Text	N	—	The reason for the Logon rejection. Only included for Logout messages in response to an invalid Logon message.	String(150)
<Standard Trailer>		Y	—	—	—

9.3 Heartbeat

A Heartbeat message is sent in response to a Test Request message to ensure connectivity.

9.3.1 Valid Responses

- None

9.3.2 Message Format

Table 11 Heartbeat Format

Tag	Field Name	Req'd	Valid Values	Description	Type
<Standard Header>		Y	—	35=0	—
112	TestReqID	C	—	Required if heartbeat message is generated in response to a Test Request message. In this case, this tag must contain the `TestReqID` that was sent in the Test Request message.	String(20)
<Standard Trailer>		Y	—	—	—

9.4 Test Request

The Test Request message forces a heartbeat from the receiving application. The receiving application responds to the Test Request message with a Heartbeat message that includes the `TestReqID` (#112) value from the received Test Request.

9.4.1 Valid Responses

- Heartbeat (#35=0)

9.4.2 Message Format

Table 12 Test Request Format

Tag	Field Name	Req'd	Valid Values	Description	Type
<Standard Header>		Y	—	35=1	—
112	TestReqID	C	—	This identifier should be returned in the Heartbeat response.	String(20)
<Standard Trailer>		Y	—	—	—

9.5 Session-level Reject

A Session-level Reject message is sent by OCX when a message is received but cannot be processed.

9.5.1 Valid Responses

- None

9.5.2 Message Format

Table 13 Session-level Reject Format

Tag	Field Name	Req'd	Valid Values	Description	Type (Length)
<Standard Header>		Y	—	35=3	—
45	RefSeqNum	Y	—	`MsgSeqNum` (#34) of the message being rejected.	int(9)
371	RefTagID	N	—	Tag ID of the field being referenced.	int(4)
372	RefMsgType	Y(OCX)	—	`MsgSeqNum` (#35) of the message being rejected.	String(2)
373	SessionRejectReason	Y(OCX)	See "Session Rejection Reasons" on page 41.	Rejection code	int(4)
58	Text	Y(OCX)	See "Session Rejection Reasons" on page 41.	Text description of rejection reason	String(150)
<Standard Trailer>		Y	—	—	—

9.6 Resend Request

The receiving application sends a Resend Request message to initiate the retransmission of messages, typically when it detects a gap in the message sequence number. The Resend Request message can request a single message or a range of messages.

9.6.1 Message Format

Table 14 Resend Request Format

Tag	Field Name	Req'd	Valid Values	Description	Type
<Standard Header>		Y	—	35=2	—
7	BeginSeqNo	Y	—	Message sequence number of the first message in range to be resent.	int(9)
16	EndSeqNo	Y	—	Message sequence number of the last message to be resent.	int(9)
<Standard Trailer>		Y	—	—	—

10 Application Message Formats

10.1 New Order - Single

Customers use the New Order - Single message to submit orders in OCX.

10.1.1 Valid Responses

- Execution Report (#35=8)
- Business Message Reject (35=j)

10.1.2 Message Format

Table 15 New Order - Single Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=D	—
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)
21	HandlInst	Y	1=Automated execution order	Instructions for handling the order	char(1)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
60	TransactTime	Y	—	Time this order was initiated by the trading system.	UTCTimestamp(21)
38	OrderQty	Y	—	Order quantity. Must be a positive number.	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	C	—	Required for limit orders `OrdType` (#40)=2	Price(20)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
15	Currency	Y	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers. The value of this tag must be between 1 and the value in `OrderQty` (#38).	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
<Standard Trailer>		Y	—	—	—

10.2 Order Cancel/Replace Request

Customers send the Order Cancel/Replace Request to request cancelation of an existing order and replacement with a new order. OCX handles this process in a single transaction from the order workflow perspective: The replacement order is not created unless and until the original order is successfully canceled.

When OCX receives an order cancel/replace request, OCX cancels the original order, replaces it with the modified order, and sends the confirmation of the modified order to the customer.

Customers can use the Order Cancel/Replace Request message to modify the following order parameters:

- Quantity (`OrderQty` (#38))
- Price (`Price` (#44))
- Side (`Side` #54))
- Maximum amount to be shown (`MaxShow` (#210))
- Peg parameters (`PegPriceType` (#1094), `PegOffsetType` (#836), `PegOffsetValue` (#211))

The Order Cancel/Replace Request message should not be used to cancel an existing order. To cancel an existing order, the Order Cancel Request message should be used.

10.2.1 Valid Responses

- Execution Report (#35=8)
- Order Cancel Reject (#35=9)
- Business Message Reject (#35=j)

10.2.2 Message Format

Table 16 Order Cancel/Replace Request Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Strength)
<Standard Header>		Y	—	35=G	—
37	OrderID	Y(OCX)	—	Unique identifier for the order assigned by OCX.	String(20)
41	OrigClOrdID	Y	—	`ClOrdID` (#11) of the order being canceled or replaced. This order should not be a filled or rejected order.	String(11)
11	ClOrdID	Y	—	Unique identifier for the cancel/replace request assigned by customer.	String(11)
21	HandlInst	Y	1=Automated execution order	Instructions for handling the order	char(1)
55	Symbol	Y(OCX)	—	Currency pair for the order. Should match the original order.	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
60	TransactTime	Y	—	Time the cancel/replace request was initiated by the customer application.	UTCTimestamp(21)

Tag	Field Name	Req'd	Valid Values	Description	Type(Strength)
38	OrderQty	Y	—	Amount for the new (replaced) order. Must be a positive number.	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order. Should match the original order.	char(1)
44	Price	C	—	Required for limit orders `OrdType` (#40)=2. This is the price of the new order that replaces the original order.	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price. Must match the original order.	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect. Must match the original order.	char(1)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
9501	IFRMEnabled	N	Y=enabled N=disabled (default)	Protects from the risk of an order being filled between the time the customer application initiates the cancel/replace request and the time the cancel/replace request is actually completed by OCX. See "In-Flight Risk Mitigation" on page 15.	Boolean(1)
<Standard Trailer>		Y	—	—	—

10.3 Order Cancel Request

The customer sends the Order Cancel Request message to request cancelation for all of the remaining quantity of an existing order. The Order Cancel/Replace Request message should be used to partially cancel (reduce) an order.

OCX accepts the request only if the order can be successfully canceled.

10.3.1 Valid Responses

- Execution Report (#35=8)
- Order Cancel Reject (#35=9)

10.3.2 Message Format

Table 17 Order Cancel Request Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Strength)
<Standard Header>		Y	—	35=F	—
41	OrigClOrdID	Y	—	`ClOrdID` (#11) of the order being canceled.	String(11)
37	OrderID	Y(OCX)	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the cancel request assigned by customer.	String(11)
55	Symbol	Y(OCX)	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
60	TransactTime	Y	—	Time the cancel request was initiated by the customer application.	UTCTimestamp(21)
<Standard Trailer>		Y	—	—	—

10.4 Order Status Request

The customer sends the Order Status Request message to request the status of a working order (for example, unfilled or partially filled).

If the order is no longer active (no longer in the book), OCX responds with an Execution Report with `OrdStatus` (#39) of "U" (Undefined).

10.4.1 Valid Responses

- Execution Report (#35=8)

10.4.2 Message Format

Table 18 Order Status Request Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=H	—
11	ClOrdID	Y	—	The `ClOrdID` of the order whose status is being requested.	String(11)
37	OrderID	Y(OCX)	—	Unique identifier for the order assigned by OCX.	String(20)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
<Standard Trailer>		Y	—	—	—

10.5 Execution Report

OCX sends an Execution Report message to:

- Confirm the receipt of an order
- Confirm changes to an order
- Relay order status information
- Relay fill information on working orders
- Reject orders

Each Execution Report contains two fields that communicate both the current state of the order (`OrdStatus` (#39)) and the purpose of the message (`ExecType` (#150).)

10.5.1 Valid Responses

- None

10.5.2 Message Formats

- "Order Create, Cancel, or Replace" on page 28
- "Order Status Response" on page 29
- "Fill" on page 31
- "Order Rejection" on page 33
- "Order Expiration" on page 35

Message Format: Order Create, Cancel, or Replace

Table 19 Execution Report Format: Create/Cancel/Replace

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=8	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)
41	OrigClOrdID	C	—	Conditionally required for response to a Cancel or Cancel/Replace request. `ClOrdID` (#11) of the previous accepted order when canceling or replacing an order.	String(11)
150	ExecType	Y	0=New 4=Canceled 5=Replaced	Describes the purpose of the execution report.	char(1)
39	OrdStatus	Y	0=New 4=Canceled 5=Replaced	Identifies the order status as accepted, canceled or replaced	char(1)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
38	OrderQty	Y	—	Amount to be traded on the order	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	N	—	Required if specified on order	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
151	LeavesQty	Y	—	Unfilled amount for the order or quantity available for further execution.	Qty(20)
14	CumQty	Y	0	Currently executed quantity for the order.	Qty(20)
6	AvgPx	Y	0	Calculated average price for all fills on this order.	Price(20)
60	TransactTime	Y	—	Time the transaction represented by this execution report occurred.	UTCTimestamp(21)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
<Standard Trailer>		Y	—	—	—

Message Format: Order Status Response

Table 20 Execution Report Format: Order Status

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=8	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
150	ExecType	Y	I=Order Status	Describes the purpose of the execution report.	char(1)
39	OrdStatus	Y	0=New 1=Partially Filled U=Undefined	Describes the current state of an order. If the order is not in the order book, then U (Undefined) is returned.	char(1)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
38	OrderQty	Y	—	Amount to be traded on the order	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	N	—	Required if specified on order	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)
151	LeavesQty	Y	—	Unfilled amount for the order or quantity available for further execution.	Qty(20)
14	CumQty	Y	—	Currently executed quantity for the order.	Qty(20)
6	AvgPx	Y	—	Calculated average price for all fills on this order.	Price(20)
60	TransactTime	Y	—	Time the order status report acknowledgment was generated.	UTCTimestamp(21)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
<Standard Trailer>		Y	—	—	—

Message Format: Fill

Table 21 Execution Report Format: Fill

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=8	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)
17	ExecID	C	—	Unique identifier of execution message as assigned by OCX. Required if `ExecType` (#150)=F (Trade).	String(20)
375	ContraBroker	C	—	Identifies the executing broker. Required if `ExecType` (#150)=F (Trade).	String(20)
150	ExecType	Y	F=Trade	Describes the purpose of the execution report.	char(1)
39	OrdStatus	Y	1=Partially Filled 2=Filled	Identifies if the order is partially filled or completely filled.	String(20)
64	FutSettDate	C	—	Trade settlement date in YYYYMMDD format.	LocalMktDate(8)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
				Required if ExecType (#150)=F (Trade)	
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
38	OrderQty	Y	—	Amount to be traded on the order	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	N	—	Required if specified on order	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)
32	LastQty	C	—	Quantity bought/sold on this (last) fill. Required if ExecType (#150)=F (Trade)	Qty(20)
31	LastPx	C	—	Price of this (last) fill. Required if ExecType (#150)=F (Trade).	Price(20)
151	LeavesQty	Y	—	Unfilled amount for the order or quantity available for further execution.	Qty(20)
14	CumQty	Y	—	Currently executed quantity for the order.	Qty(20)
6	AvgPx	Y	—	Calculated average price for all fills on this order.	Price(20)
75	TradeDate	C	—	Indicates the business date for the trade referenced in this message. Required if `ExecType` (#150)=F (Trade).	LocalMktDate(8)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
60	TransactTime	Y	—	Time the transaction represented by this execution report occurred.	UTCTimestamp(21)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
9502	ClientTagID	N	—	Unique identifier for the customer on the other side of the execution.	String(10)
<Standard Trailer>		Y	—	—	—

Message Format: Order Rejection

Table 22 Execution Report Format: Order Rejection

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=8	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)
150	ExecType	Y	8=Rejected	Describes the purpose of the execution report.	char(1)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
39	OrdStatus	Y	8=Rejected	Describes the current state of an order.	String(20)
103	OrdRejReason	C	See Appendix A, "Business Rejection Reasons" on page 39.	Rejection code. Required when `ExecType` (#150)=8 (for example, when an order is rejected).	String(4)
58	Text	C	See Appendix A, "Business Rejection Reasons" on page 39.	Rejection reason. Required when `ExecType` (#150)=8 (for example, when an order is rejected).	String(150)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
38	OrderQty	Y	—	Amount to be traded on the order	Qty(20)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	N	—	Required if specified on order	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)
151	LeavesQty	Y	0 (zero)	Quantity available for further execution. Always 0 (zero) for rejected orders.	Qty(20)
14	CumQty	Y	0	Currently executed quantity for the order.	Qty(20)
6	AvgPx	Y	0	Calculated average price for all fills on this order.	Price(20)
60	TransactTime	Y	—	Time the transaction represented by this execution report occurred.	UTCTimestamp(21)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
<Standard Trailer>		Y	—	—	—

Message Format: Order Expiration

Table 23 Execution Report Format: Order Expiration

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=8	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX.	String(20)
11	ClOrdID	Y	—	Unique identifier for the order assigned by customer.	String(11)
150	ExecType	Y	C=Expired	Describes the purpose of the execution report.	char(1)
39	OrdStatus	Y	C=Expired	Describes the current state of an order.	String(20)
55	Symbol	Y	—	Currency pair for the order	String(7)
54	Side	Y	1=Buy 2=Sell	1 (Buy): The customer buys `Currency` (#15). 2 (Sell): The customer sells `Currency` (#15).	char(1)
38	OrderQty	Y	—	Amount to be traded on the order	Qty(20)

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
40	OrdType	Y	2=Limit P=Pegged	Type of order	char(1)
44	Price	N	—	Required if specified on order	Price(20)
15	Currency	Y(OCX)	—	Identifies currency used for price	Currency(3)
59	TimeInForce	Y	0=DAY 1=Good Till Cancel (GTC)	Specifies how long the order remains in effect	char(1)
151	LeavesQty	Y	0 (zero)	Quantity available for further execution. Always 0 (zero) for expired orders.	Qty(20)
14	CumQty	Y	—	Currently executed quantity for the order.	Qty(20)
6	AvgPx	Y	—	Calculated average price for all fills on this order.	Price(20)
60	TransactTime	Y	—	Time the transaction represented by this execution report occurred.	UTCTimestamp(21)
210	MaxShow	N	—	Maximum quantity of the order shown to other customers. If this field is not specified, then the entire order amount is shown to other customers.	Qty(20)
1094	PegPriceType	C	2=Mid-price peg	Required for pegged orders. Defines the type of pegged order.	int(1)
836	PegOffsetType	C	2=Ticks	Required for pegged orders. Specifies the type of peg offset value.	int(1)
211	PegOffsetValue	C	—	Amount (signed) in pips added to the peg for a pegged order in the context of the `PegOffsetType` (#836)	float(10)
<Standard Trailer>		Y	—	—	—

10.6 Order Cancel Reject

OCX sends the Order Cancel Reject message to notify the customer application of a rejected Order Cancel Request or Order Cancel/Replace Request message.

10.6.1 Valid Responses

- None

10.6.2 Message Format

Table 24 Order Cancel Reject Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=9	—
37	OrderID	Y	—	Unique identifier for the order assigned by OCX. If `CxlRejReason` (#102) = Unknown Order, this value will be set to NONE.	String(20)
11	ClOrdID	Y	—	Unique identifier for the cancel or replace request which is being rejected.	String(11)
41	OrigClOrdID	Y	—	`ClOrdID` of the order that could not be canceled/replaced.	String(11)
39	OrdStatus	Y	U=Undefined	Order status is undefined.	char(1)
58	Text	Y	See Appendix A, "Business Rejection Reasons" on page 39.	The reason code for the Order Cancel rejection.	String(150)
434	CxlRejResponseTo	Y	1=Order Cancel Request 2=Order Cancel/Replace Request	Identifies the type of request the cancel is in response to.	char(1)
102	CxlRejReason	Y	See Appendix A, "Business Rejection Reasons" on page 39.	Code to identify reason for cancel rejection.	int(1)
<Standard Trailer>		Y	—	—	—

10.7 Business Message Reject

OCX sends the Business Message Reject message to notify the customer application of a rejected New Order - Single or Order Cancel/Replace Request message.

10.7.1 Valid Responses

- None

10.7.2 Message Format

Table 25 Business Message Reject Format

Tag	Field Name	Req'd	Valid Values	Description	Type(Length)
<Standard Header>		Y	—	35=j	—
372	RefMsgType	Y	D (New Order -Single) G (Order Cancel/Replace Request)	The `MsgType` (#35) of the message being referenced	String(2)
379	BusinessRejectRefID	Y	—	The `ClOrdID` (#11) of the rejected message	String(11)
380	BusinessRejectReason	Y	See Appendix A, "Business Rejection Reasons" on page 39.	Reason code	String(11)
58	Text	Y	See Appendix A, "Business Rejection Reasons" on page 39.	Text description of rejection reason	String(150)
<Standard Trailer>		Y	—	—	—

APPENDIX A

Business Rejection Reasons

The following table lists the business rejection reasons returned by OCX after a message passes session-level validation and is accepted for processing. For the list of session-level rejection reasons, see "Session Rejection Reasons" on page 39.

Table 26 Business Rejection Reasons

Error Code	Text Explanation of Error Code
5	Conditionally required field missing: • `Price` (#44) is required for a limit order • `PegPriceType` (#1094) is required for a peg order • `PegOffsetType` (#836) is required for a peg order • `PegOffsetValue` (#211) is required for a peg order
1001	Currency pair is not supported
1002	Orders may not be entered while the venue is closed
1003	The order is not in the book
1004	Order quantity is outside of the allowable range
1005	Order price is outside of the allowable range
1006	The user is not authorized to trade
1007	Time In Force is not supported
1008	Term currency orders are not supported
1009	Price for limit orders must be greater than 0
1010	Price is required for limit orders
1011	Order type is not supported
1012	Side is not supported
1013	Display quantity cannot be more than the order quantity
1014	Peg price type is not supported
1015	Peg offset type is not supported
1016	Peg price type, peg offset type and peg offset are required for pegged orders
1017	`ClOrdID` (#11) and `OrderID` (#37) of the order being canceled or replaced do not match

Error Code	Text Explanation of Error Code
1018	Pegged orders cannot be accepted while the peg reference price is not available
1019	Order quantity exceeds the available credit
1021	Maximum length of `ClOrdID` (#11) should be 11.
1022	Maximum length of `OrigClOrdID` (#41) should be 11.
1023	`OrderID` (#37) should contain only digits and length should be less than 20.

APPENDIX B

Session Rejection Reasons

The following table lists the rejection reasons returned by OCX in a Session-level Reject message when a message fails session-level validation. For the list of business rejection reasons sent by OCX in Business Message Reject (#35=j), Execution Report (#35=8), Order Cancel Reject (#35=9) messages, see "Business Rejection Reasons" on page 39.

Table 27 Session Rejection Reasons

Error Code	Text Explanation of Error Code
0	Invalid tag number
1	Required tag missing
2	Tag not defined for message type
3	Undefined Tag
4	Tag specified without a value
5	Value is incorrect (out of range) for this tag
6	Incorrect data format for value
7	Decryption problem
8	Signature problem
9	CompID problem
10	`SendingTime` (#52) accuracy problem
11	Invalid `MsgType` (#35)
12	XML Validation error
13	Tag appears more than once
14	Tag specified out of required order
15	Repeating group fields out of order
16	Incorrect number in group count for repeating group
17	Non "data" value includes field delimiter (SOH character)
99	Other
1020	Volume control message violations. See "Messaging Controls" on page 16.

APPENDIX C

Changes

The following tables lists API changes and significant changes to this document in chronological order from newest to oldest.

Table 28 API and Document Changes

Date (Document Version)	Changes

September 2015 (1.0v3)	• Added "Business Message Reject" on page 37
	• Updated "Messaging Thresholds" on page 16
	• Renamed appendix "Order Rejection Reasons" to "Business Rejection Reasons"
	• Added error codes 5, 1021, 1022, and 1023 to appendix "Business Rejection Reasons" on page 39
	• Added appendix "Session Rejection Reasons" on page 41
	• Added `Text` (#58) to the following messages:
	  ○ "Session-level Reject" on page 20
	  ○ "Execution Report Format: Order Rejection" on page 33
	  ○ "Order Cancel Reject" on page 36
	• Values changed in "Execution Report Format: Create/Cancel/Replace" on page 28:
	  ○ `CumQty` (#14)=0 (zero)
	  ○ `AvgPx` (#6) =0 (zero)
	• Added `ContraBroker` (#375) field to "Execution Report Format: Fill" on page 31