FXInside™

FIX Client API Guide

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

Friday, June 26, 2015 5.6v1

Contents

Foreword

About This Document

This document describes the FIX Client API that FX Grid market participants can use to access FX Grid trading functionality.

About FX Grid

FX Grid offers access to a global real-time system of liquidity providers and liquidity takers that provides the most robust and lowest cost means of electronically connecting leading market participants amongst each other for dynamic and customizable outsourcing and in-sourcing of FX and interest rate liquidity. Market participants connect to the network through a single interface (API) from which they can interact with other participants for negotiation, execution, and settlement of trades. FX Grid provides a single point of liquidity to communicate with other market participants, by providing adaptors that translate each participant's communications messages into the message format of the recipient.

Organization of This Document

The following chapters discuss the configuration of the FIX Client API in general and the supported trading workflows in specific:

- "FIX Solution Overview" on page 10
- "Session Management" on page 65
- "Market Data and Quote Workflow" on page 70
- "Trading Workflow" on page 89

In addition, the appendix "Changes" on page 156 lists the revisions made to the content of this document.

Typographic Conventions

This document presents information with consistent conventions to make the information easy to understand and use.

Italic	■ Emphasize a term or concept ■ Introduce new terms. For example, *system monitoring*.
Bold	User-interface elements. For example, **File** menu.

`Sans Serif`	■ Names of classes, instances, messages, and examples of code. For example, `Counterparty`. ■ Filenames, pathnames, commands, and other operating-system constructs. For example, `/cust/usr`.
`Italic Sans Serif`	Variable elements for which you must substitute a value. For example, *`filename`*.java.
Blue color	URLs and cross-references that you can click when viewing the document online. For example, "Typographic Conventions" on page 7.
`ItalicNameRoot``/bin/util/`	Root path references. For example, *`ProductRoot`* (to refer to the root path `/cust/usr/Application`) In file names, a directory name that ends with "Root" is a variable that represents a root path. To clarify that it is a variable and not an actual path name, the name appears in italics.

`/root/xmlStructure/example@data='test'`	References to elements in an XML doc are made using XPath conventions. The example reads: "The `example` element with a `data` attribute value of `'test'` in the `xmlStructure` in the `root` element."
	The XML equivalent would be:
	`<root>`
	`<xmlStructure>`
	`<example data="test"/>`
	`</xmlStructure>`
	`</root>`

This document uses the following symbols and conventions to designate certain items or relationships.

***Table 4-1** Structural Conventions*

Format	Description
`OrdStatus` (#39)	The FIX field name with the FIX field number in parentheses
`PathnameRoot`	Root path references In file names, a directory name that ends with "Root" is a variable representing a root path that depends on your installation. For example: *`integralProductRoot`*`/broker/appLogs/` could refer to the path: `/cust/usr/integral/broker/appLogs/`
`singleSourceCodeElement WrappedToTheNextLine`	A long element name or line of code that is wrapped and indented to fit a cell, column, or page For example: `anObjectWithVeryLongName. methodOnObject` To avoid confusion between different programming languages, this document does not use special symbols to identify the break unless the programming language provides such a mechanism.

CHAPTER 1

FIX Solution Overview

1.1 Introduction

This document describes the FX Inside FIX Client API, a FIX-based channel that allows clients to deal with liquidity providers offering foreign exchange liquidity through FX Grid.

1.2 Business Rules

The following sections describe how the FIX Client API applies the general FIX protocol to satisfy your business needs.

1.2.1 Rates To Discard

There are several cases when you should discard a rate as these are situations under which the rate is inactive or is not tradable:

On the Market Data Snapshot/Full Refresh message:

- `QuoteCondition` (#276)=B (Closed/Inactive)

- `MDEntryPx` (#270)=0 (not tradable)
- `MDEntrySize` (#271)=0 (not tradable)

On the Quote message, if any of the following are included and have a value of 0 (zero):

- `BidSpotRate` (#188)
- `OfferSpotRate` (#190)
- `BidPx` (#132)
- `OfferPx` (#133)
- `BidSize` (#134)
- `OfferSize` (#135)

In each of these cases, your client should ignore the incoming quote.

1.2.2 Sessions

The FIX Client API distinguishes between two session types to optimize your trading message flows:

- Quote: Session for sending/receiving quotes. Messages are time-sensitive and transient to enable the high message volume typically associated with quotes and quote streams. The server does not resend quote session messages in response to a resend request from the client. You must establish a quote session to initiate all trading workflows. See "Trading Workflows" on page 12.
- Order: Session for order submission and trade execution. Messages are transactional and persistent with no lost messages allowed, reflecting their business criticality. The server resends order session messages in response to a resend request from the client. How you use an order session depends on the trading workflow you employ:
  - Executable Streaming Prices (ESP) workflow: You establish the order session to hit quotes received in the form of market data messages on the quote session. New Order – Single messages sent in this session refer to the `QuoteEntryID` (#299) value on a quote and have an `OrdType` (#40) of `D` for "previously quoted". See "Executable Streaming Prices (ESP) Workflow" on page 12.
  - Order workflow: You establish the order session to submit, replace, and cancel limit orders. New Order – Single messages sent in this session have an `OrdType` (#40) of `2` for "limit". See "Order Workflow" on page 13.
  - Request for Stream (RFS) workflow: You establish a quote session to request and receive quotes from a provider and then use the order session to hit the quotes. New Order – Single messages sent in this session refer to the `QuoteEntryID` (#299) value on a quote and have an `OrdType` (#40) of `D` for "previously quoted". See "Request for Stream (RFS) Workflow" on page 17.

You must establish each FIX session separately with the server. You use the same credentials for each session type. You can have multiple FIX sessions under one server connection. You do not have to reconnect for each session.

The ID that you set for your organization ID on messages includes an indicator of the session type. See "Your Organization Client ID" on page 54.

1.2.3 Trading Workflows

The FIX protocol defines the messages for a general representation of trading workflow (request > quote > order > execution > post-trade).

The FIX Client API extends this general approach to model specific trading workflows that more closely match your business requirements.

- "Request/Quote Workflows" on page 12
- "Post-Order Workflows" on page 19
- "Post-Trade Workflows" on page 22

Request/Quote Workflows

(FIX workflow: **request > quote > order** > execution > post-trade)

The FIX Client API currently supports the following request/quote workflows:

- "Executable Streaming Prices (ESP) Workflow" on page 12
- "Order Workflow" on page 13
- "Order Amend and Cancel/Replace Workflow" on page 16
- "Request for Stream (RFS) Workflow" on page 17
- "Batch Trade Workflow" on page 18

These workflow examples have been simplified by assuming successful post-order trade execution. For the details of possible trading workflows after order submission, See "Post-Order Workflows" on page 19.

Executable Streaming Prices (ESP) Workflow

The Executable Streaming Prices (ESP) workflow involves the quote and order sessions. For more information about sessions, see "Sessions" on page 11.

The client requests market data in a quote session, receives executable quotes on the same quote session, and then in an order session sends orders that refer to a quote's `QuoteEntryID` (#299).

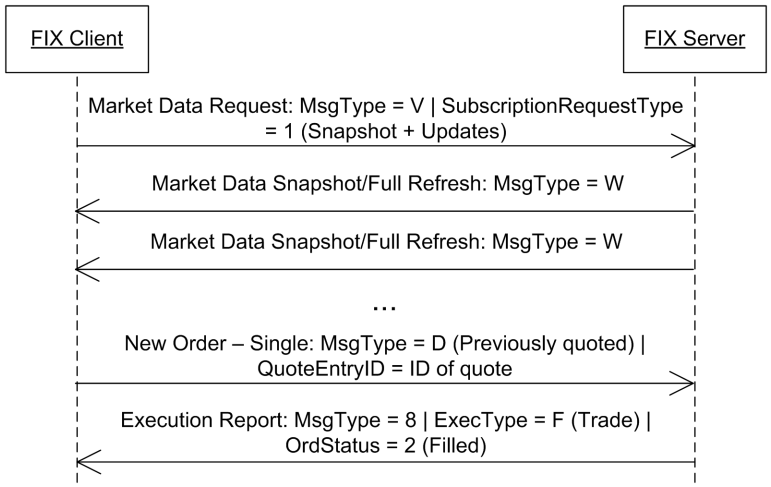

***Figure 1-1** Typical ESP Trading Workflow*

The following trading messages are applicable to the ESP workflow:

***Table 1-1** ESP Trading Messages*

Message (Direction)	Session	Comments
"Market Data Request" on page 71	Quote	Client requests ESP
"Market Data Snapshot/Full Refresh" on page 75	Quote	Server sends prices for the requested currency pairs
"Market Data Request Reject" on page 77	Quote	Server rejects the market data request (for example, the currency pair is not supported)
"New Order – Single" on page 90	Order	Client submits an order to the server in response to a price received in a Market Data Snapshot/Full Refresh message
"Execution Report" on page 121	Order	Server sends the current order status to the FIX client. An Execution Report with `ExecType` (#150) value 0 (New) is optional and may be skipped. See "Order Status" on page 36 for details.

Order Workflow

The Order workflow involves the order session and, optionally, the quote session. The client requests market data for price discovery only on a quote session. Price discovery is an optional part of the order workflow. The client sends orders on an order session. For more information about sessions, see "Sessions" on page 11.

The value of the `ExecInst` (#18) field on the order message determines how the order is executed, for example whether the order is crossed with quotes on the server or broadcast to and lifted by other market participants. For more information about order execution options, see "Order Execution" on page 30. For information about the field, see "ExecInst" on page 92.

After the client submits an order, the server checks the order's validity. The server may then acknowledge the order with a "New" Execution Report message (`ExecType` (#150)=0). If the order can be filled immediately, the server may skip the "New" ack message and send only a "Trade" Execution Report message (`ExecType` (#150)=F). If your organization's workflow requires order acknowledgement with the "New" Execution Report message (`ExecType` (#150)=0), please contact your Integral Solutions Manager.

The basic workflow for all order types is the same with minor differences. For stop and stop limit orders, the server sends an additional Execution Report message with `WorkingIndicator` (#636) set to "Y" when the order is triggered.

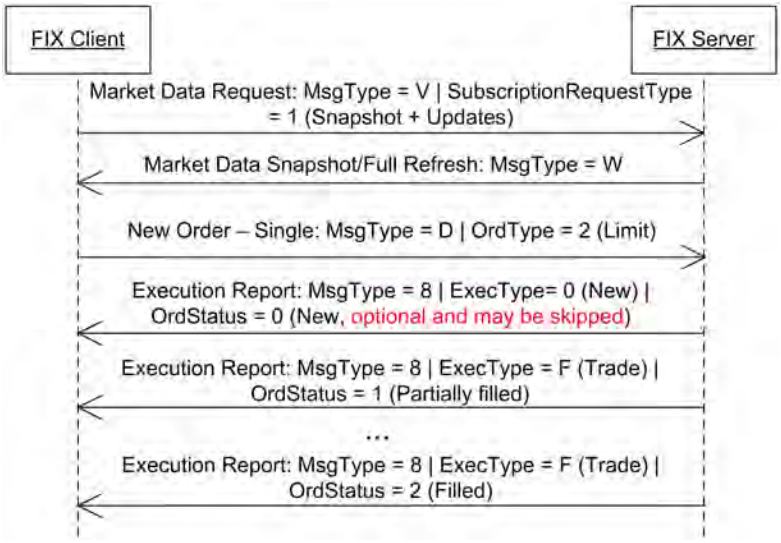

Figure 1-2 *Order Trading Workflow (Limit Orders)*

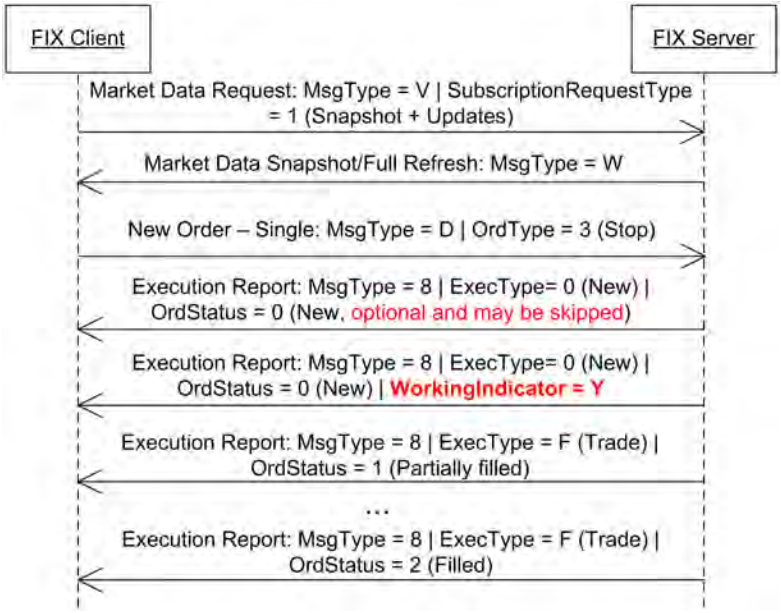

Figure 1-3 *Order Trading Workflow (Stop and Stop Limit Orders)*

Figure 1-4 *Order Trading Workflow (Algo Order with Delayed Start)*

Table 1-2 *Order Trading Messages*

Message (Direction)	Session	Comments
"New Order – Single" on page 90	Order	Client submits an unsolicited order to the server
"Execution Report" on page 121	Order	Server sends the current order status to the FIX client. The "New" Execution Report (`ExecType` (#150)=0) is optional and may be skipped. See "Order Status" on page 36 for details. When a stop or stop limit order has been triggered, the server sends an Execution Report with `WorkingIndicator` (#636) set to "Y" (see "WorkingIndicator" on page 129. An Execution Report with `OrdStatus` (#39)=7 (Stopped) is sent for new algo orders with a delayed start and for orders that have been suspended by the trading user.
"Order Cancel Request" on page 105	Order	Client cancels an order that was previously submitted to the server
"Order Cancel/Replace Request" on page 106	Order	Client cancels an existing order and replaces it with a new order. See "Order Amend and Cancel/Replace Workflow" on page 16.
"Order Cancel Reject" on page 115	Order	Server rejects the client's request to cancel an order. The `CxlRejResponseTo` (#434) field is set to 1=Order Cancel Request.
"Order Mass Cancel Request" on page 117	Order	Client cancels all open orders on sever
"Order Mass Cancel Report" on page 118	Order	Server responds to client's order mass cancel request

***Table 1-2** Order Trading Messages (continued)*

Message (Direction)	Session	Comments
"Order Status Request" on page 119	Order	Client requests the current status of a specific order
"Order Mass Status Request" on page 120	Order	Client requests the current statuses of all open order
"Business Message Reject" on page 133	Order	Server sends to reject an order status request if the order does not exist

Order Amend and Cancel/Replace Workflow

NOTE: Depending on how your system is configured and on the state of the order, the original order may be cancelled or amended. See "Order Cancel/Replace and Amendment" on page 28 for more information.

The Order Cancel/Replace workflow involves the order session. The client sends the cancel/replace request on an order session. For more information about sessions, see "Sessions" on page 11.

After the client submits the request, the server returns two messages for a successful cancel/replace:

1. The server sends a "Pending Replace" message to confirm receipt of the request.
2. After the original order has been cancelled or amended, the FIX server sends a "Replaced" message that confirms the successful replacement or amendment. The "Replaced" message contains the details of the new order, including the order ID (`OrderID` (#37)) if the order was replaced instead of being amended.

For the continued workflow of the new order, see "Order Workflow" on page 13.

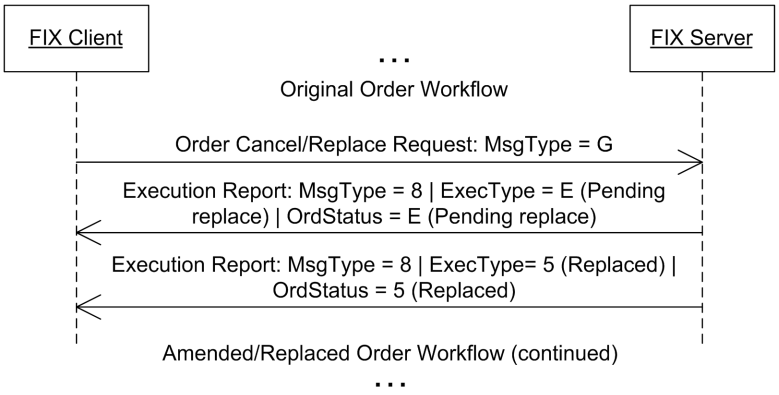

***Figure 1-5** Order Cancel/Replace Trading Workflow*

***Table 1-3** Order Cancel/Replace Trading Messages*

Message (Direction)	Session	Comments
"Order Cancel/Replace Request" on page 106	Order	Client submits request to cancel and replace an existing order to the server
"Execution Report" on page 121	Order	Server sends the status of both the original order and the new or amended order

Table 1-3 *Order Cancel/Replace Trading Messages (continued)*

Message (Direction)	Session	Comments
"Order Cancel Reject" on page 115	Order	Server rejects the client's request to cancel an order. The `CxlRejResponseTo` (#434) field is set to 2=Order Cancel/Replace Request.

Request for Stream (RFS) Workflow

The RFS workflow involves the quote and order sessions. For more information about sessions, see "Sessions" on page 11.

The client requests a quote on a quote session and receives executable quotes the same quote session, and then in an order session sends orders that refer to a quote's `QuoteEntryID` (#299).

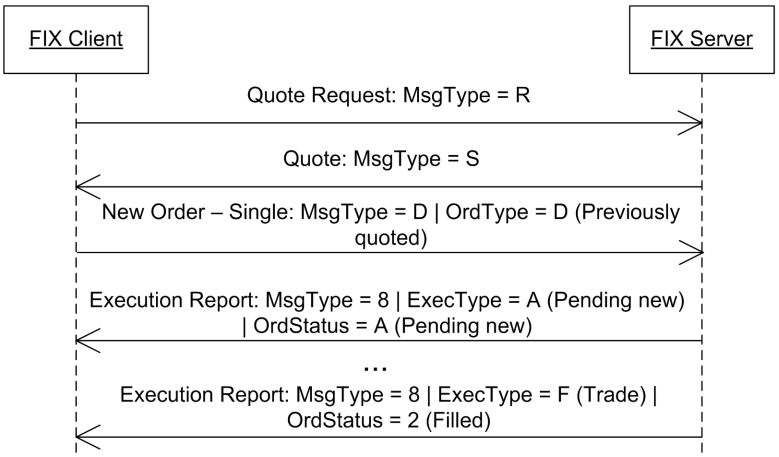

Figure 1-6 *RFS Trading Workflow*

Table 1-4 *RFS Trading Messages*

Message (Direction)	Session	Comments
"Quote Request" on page 79	Quote	Client requests streaming quotes from the server
"Quote Request Reject" on page 82	Quote	Server rejects client request for quotes
"Quote" on page 83	Quote	Server sends a quote to the FIX client. The quote can be streaming. Subsequent quotes override the previous quote.
"Quote Cancel" on page 87	Quote	■ Server cancels the quote ■ Client cancels a Quote Request
"New Order – Single" on page 90	Order	Client submits an order to the server in response to a quote
"Execution Report" on page 121	Order	Server sends the current order status to the FIX client. An Execution Report with `ExecType` (#150) value 0 (New) is optional and may be skipped. See "Order Status" on page 36 for details.

Batch Trade Workflow

The batch trading workflow involves the quote and order sessions.

NOTE: The batch trade workflow is only available to clients using the FIX 4.4 protocol.

The client requests market data in a quote session, receives executable quotes on the same quote session, and then in an order session sends orders that refer to a quote's `QuoteEntryID` (#299). The only messages with values specific to batch trades are the Quote Request and the resulting Quote.

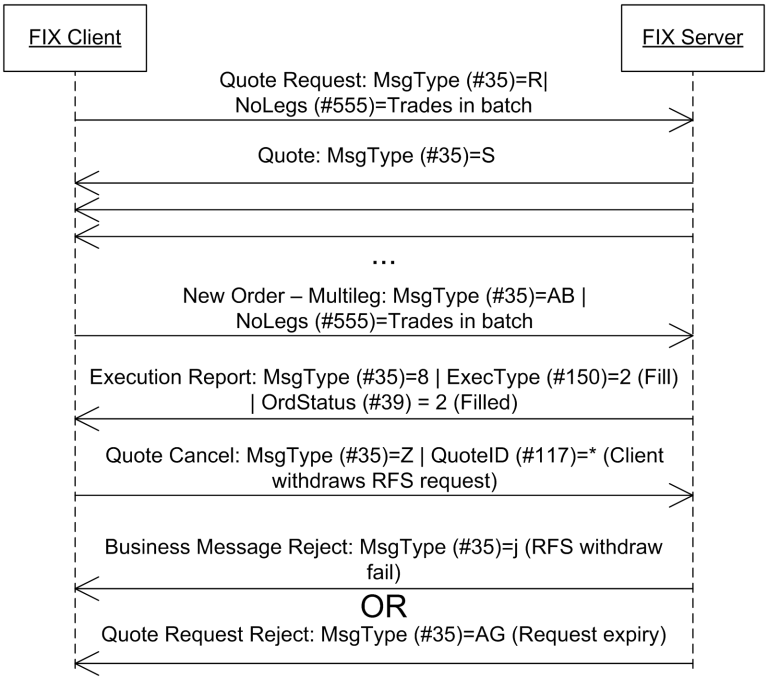

***Figure 1-7** Typical Batch Trade Workflow*

The following trading messages are applicable to the batch trade workflow:

***Table 1-5** RFS Trading Messages*

Message (Direction)	Session	Comments
"Quote Request" on page 79	Quote	Client requests streaming quotes for a batch of trades from the server
"Quote Request Reject" on page 82	Quote	Server rejects client request for quotes or the request expires
"Quote" on page 83	Quote	Server sends a quote to the FIX client. The quote can be streaming. Subsequent quotes override the previous quote.
"Quote Cancel" on page 87	Quote	■ Server cancels the quote ■ Client cancels a Quote Request
"New Order – Multileg" on page 103	Order	Client submits a batch order to the server in response to a quote

Table 1-5 *RFS Trading Messages (continued)*

Message (Direction)	Session	Comments
"Execution Report" on page 121	Order	Server sends the current order status to the FIX client. For batch trades, the valid values of `ExecType` (#150) and `OrdStatus` (#39) are 2 (Fill) and 8 (Rejected). See "Order Status" on page 36 for details.

Post-Order Workflows

(FIX workflow: request > quote > **order > execution** > post-trade)

After you submit your order, FIX Client API represents the possible outcomes with the following post-order workflows:

- "FIX Rejection" on page 19
- "Order Submission Failure" on page 19
- "Trade Done/Verified (Single Fill)" on page 21
- "Partial Fill with Unfilled Amount Canceled" on page 22

FIX Rejection

If your New Order – Single message contains invalid FIX values or is not formatted correctly, the FIX server rejects your message as invalid and sends a Business Message Reject message (see "Business Message Reject" on page 133).

IMPORTANT: If the `Text` (#58) field of any message includes the text "INTERNAL_SERVER_ERROR", this indicates a serious error. Stop trading immediately and contact Business Support to resolve the issue.

Figure 1-8 *FIX Rejection*

Order Submission Failure

If your order submission fails because of a connection error or a business reason, such as an invalid price or a failed credit check, the FIX server responds with an Execution Report message for either an order cancel or an order reject depending on the reason for the failure.

IMPORTANT: If the `Text` (#58) field of any message includes the text "INTERNAL_SERVER_ERROR", this indicates a serious error. Stop trading immediately and contact Business Support to resolve the issue.

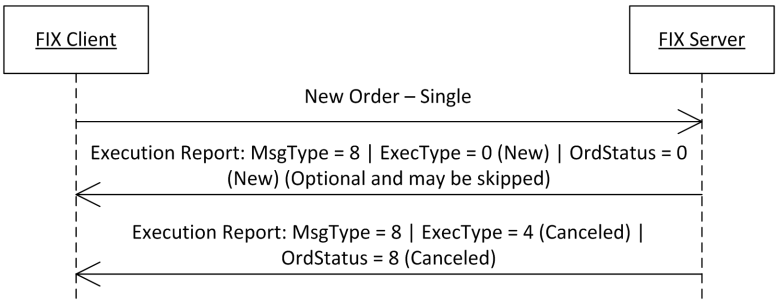

Figure 1-9 *Order Submission Failure: Cancellation*

The FX Grid sends an order cancel message for the following reasons:

- Credit check failure
- Liquidity regeneration is in progress
- Currency pair not supported
- Trading channel not supported
- Order has expired before receipt (immediate expiry or very short expiry time)

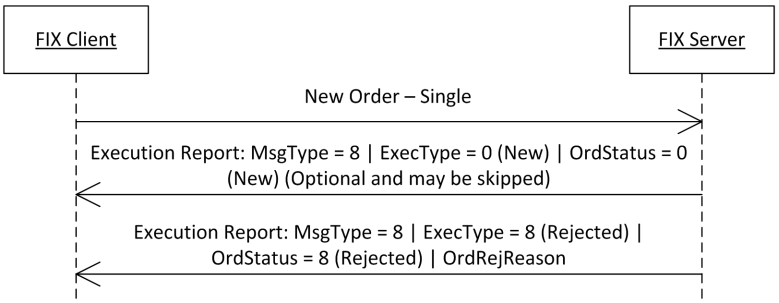

Figure 1-10 *Order Submission Failure: Rejection*

The FX Grid sends an order reject message for the following reasons:

- Legal entity mismatch
- Invalid order size
- Invalid order price
- Currency pair mismatch (invalid dealt currency)
- Incorrect/invalid values for `OnBehalfOfCompID` (#115) or `OnBehalfOfSubID` (#116)
- Incorrect/invalid value `SenderSubID` (#50)
- Missing stop price for a stop order
- Missing trading relationship in the sales dealer workflow (when trading on behalf of another organization)
- Liquidity rules validation failure (for example, the order size is greater than the maximum order size allowed)
- For algo orders, both incompatible slice sizes are specified: regular slice size (`AlgoParameters` (#7560)=SRS) and slice size as a percentage of the top-of-book (`AlgoParameters` (#7560) =TOBP)
- Incorrect peg type

If an order submission fails with a reject message, the reason for the failure, if any, is set in the `OrdRejReason` (#103) and `Text` (#58) fields of the Execution Report message.

> **IMPORTANT:** If the `Text` (#58) field of any message includes the text "INTERNAL_SERVER_ERROR", this indicates a serious error. Stop trading immediately and contact Business Support to resolve the issue.

See "OrdRejReason" on page 126 for the possible values of the `OrdRejReason` (#103) field or consult the FIX protocol documentation for your FIX version.

Trade Done/Verified (Single Fill)

If your order is accepted and executed, the FIX server sends an Execution Report message with the details of the trade. If your order is filled with a single trade, the FIX server sends a single Execution Report message. The Execution Report with `ExecType` (#150) value 0 (New) is optional and may be skipped. See "Order Status" on page 36 for details.

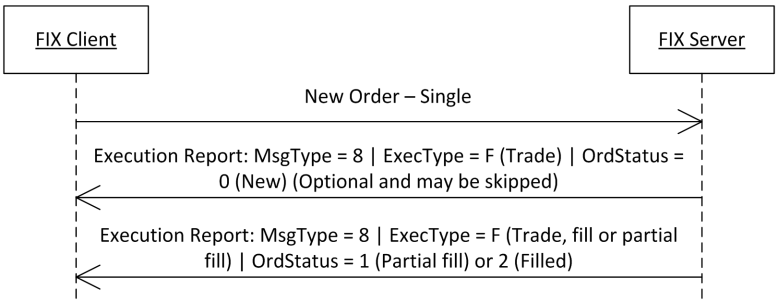

Figure 1-11 *Trade Done/Verified (Single Fill)*

Trade Done/Verified (Multiple Fills)

If your order allows multiple fills by setting the `MinQty` (#110) value to zero or less than `OrderQty` (#38), then the FIX server can send multiple Execution Report messages. Use `LastPx` (#31) to determine the price for an individual fill or `AvgPx` (#6) for the average price of all fills on the order up to the latest fill. The Execution Report with `ExecType` (#150) value 0 (New) is optional and may be skipped. See "Order Status" on page 36 for details.

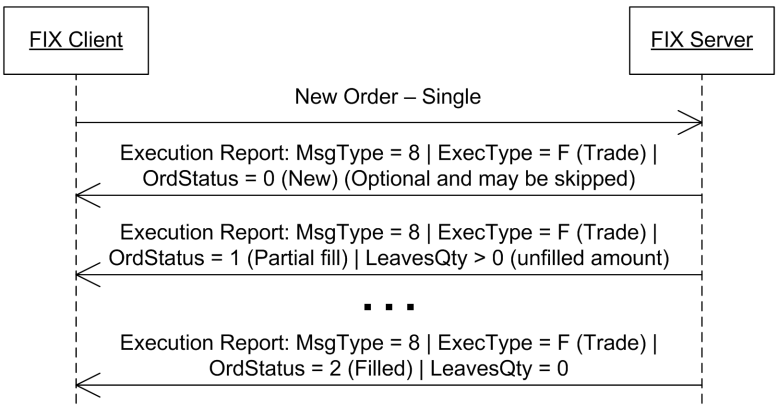

Figure 1-12 *Trade Done/Verified (Multiple Fills)*

Partial Fill with Unfilled Amount Canceled

Your originating New Order – Single request allows partial fills (see "Partial Fills" on page 32) and has a time in force of Immediate or Cancel (IOC, see "Order Expiry" on page 31). If execution results in a partial fill, the remaining amount of your order is canceled. The server sends two Execution Report messages:

- The partially filled amount with `ExecType` (#150) value of "F" (trade, partial fill or fill) `OrdStatus` (#39) with a value of "1" (partially filled).
- The canceled amount with an `ExecType` (#150) value of "4" (canceled) and `OrdStatus` (#39) with a value of "4" (canceled)

In both Execution Report messages, the `LeavesQty` (#151) field is the unfilled amount and the `CumQty` (#14) fields is the amount done.

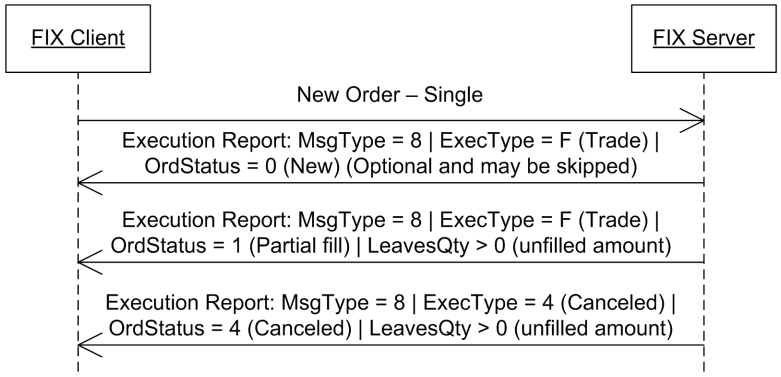

Figure 1-13 *Partial Fill with Unfilled Amount Canceled (IOC Workflow)*

Post-Trade Workflows

(FIX workflow: request > quote > order > execution > **post-trade**)

The FIX Client API includes messages for post-trade activity, such as STP download and trade status query.

STP Download

You can choose to receive STP download via FIX. When a trade is done, a Trade Capture Report is sent to the FIX client asynchronously.

If a FIX session is not available when the trade is done, the trade message is persisted on the server side. When a FIX session is re-established, Trade Capture Reports are issued to the FIX client.

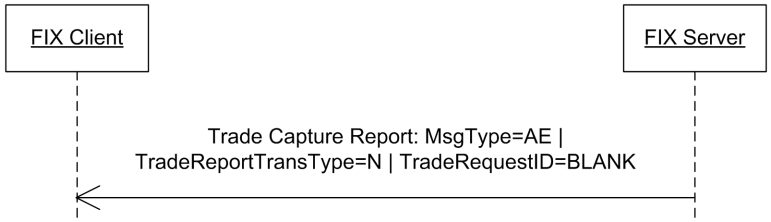

Figure 1-14 *STP Download Workflow*

STP Resend

If you receive STP download via FIX, STP resends can be triggered by system administrators. The workflow is the same as for STP download: a Trade Capture Report is sent to the FIX client asynchronously.

If a FIX session is not available when the trade is done, the trade message is persisted on the server side. When a FIX session is re-established, Trade Capture Reports are issued to the FIX client.

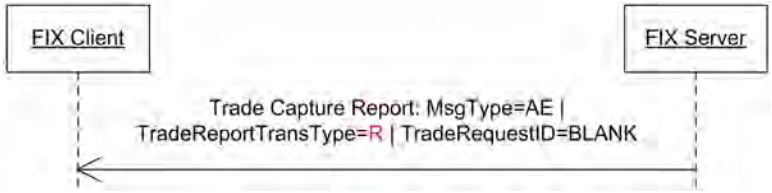

Figure 1-15 *STP Resend Workflow*

Trade Status Query

The FIX client can query the status of trades by providing a specific trade ID or a date/time range in which trades were done. The FIX server responds with a series of Trade Capture Reports for each requested trades.

A status query returns all trades that match the request criteria, not just trades that have been initiated from the FIX session.

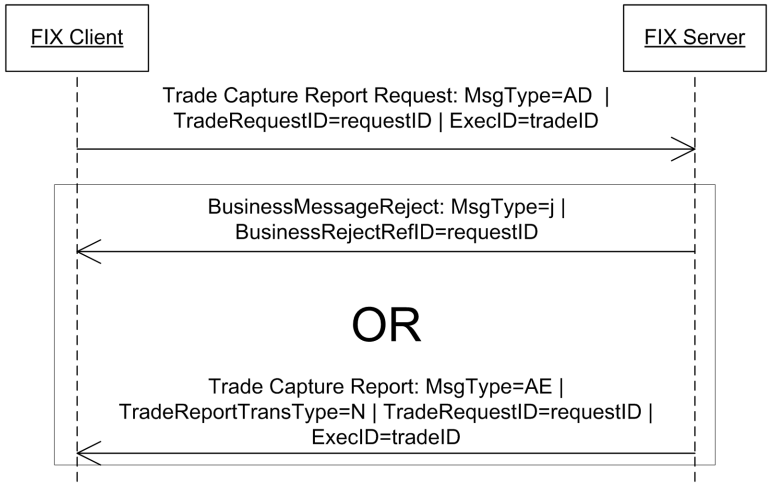

Figure 1-16 *Trade Status Query (Trade ID) Workflow*

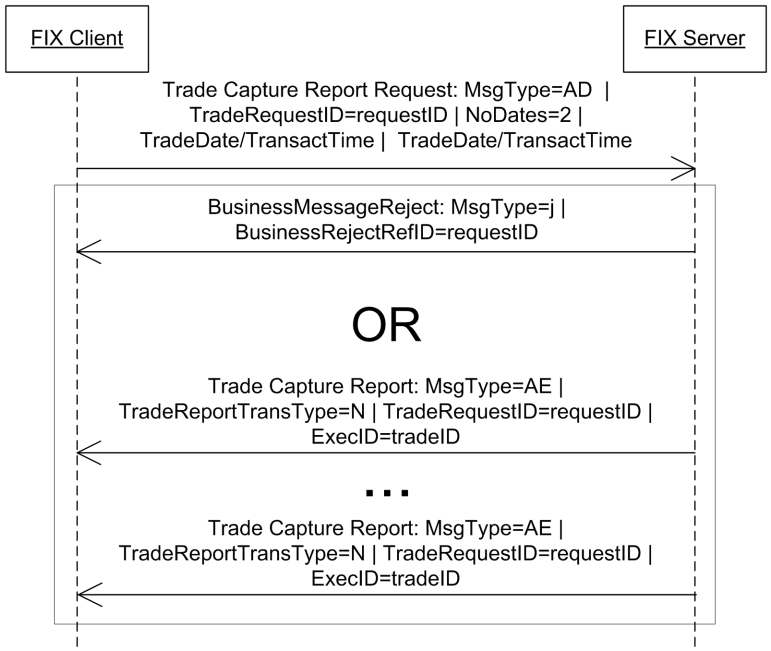

Figure 1-17 *Trade Status Query (Date Range) Workflow*

Positions Management

The FIX client can query the status of your open and settled positions. A broker can also request a customer's positions with liquidity providers. Positions can be sent as a snapshot as trades are executed.

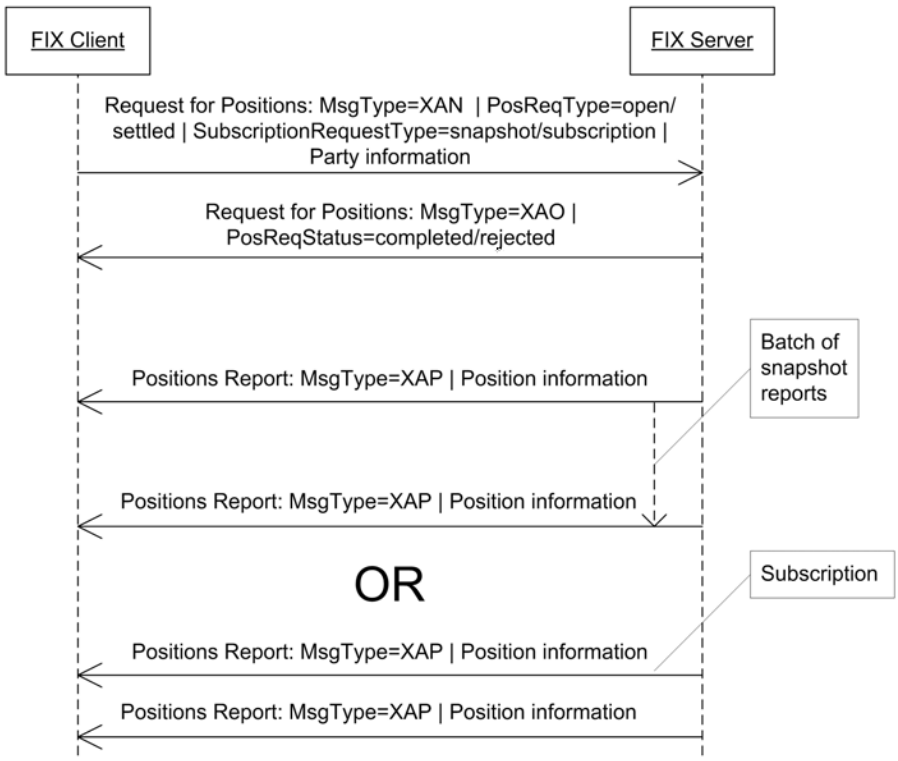

Figure 1-18 *Positions Workflow*

Applicable Messages

The following messages are used for post-trade workflows:

Table 1-6 *Post-Trade Messages*

Message (Direction)	Session	Comments
"Trade Capture Report Request" on page 138	Order	Client requests trade status with a specific trade ID or a date/time range
"Trade Capture Report" on page 139	Order	Server sends trade status. This can be triggered by: ■ STP Download ■ STP Resend ■ Trade Capture Report Request
"Business Message Reject" on page 133	Quote	Server rejects an invalid Trade Capture Report Request
"Request for Positions" on page 134	Order	Client requests position information by counterparty, currency pair, position status (open/settled), by date range for open positions, and by request type (snapshot/snapshot+subscription)
"Request for Positions Ack" on page 135	Order	Server sends acknowledgement indicating success or failure of the request
"Positions Report" on page 136	Order	Server sends a snapshot of position information as a batch of messages, one Positions Report message per currency pair. If the client requested snapshot+subscriptions, the server sends additional unsolicited Positions Report messages for updates as trades are executed.

1.2.4 Supported Trade Types

The FIX Client API currently supports the following trade types:

- ESP workflow: FX spot only
- Orders workflow: FX spot only
- RFS workflow:
  - FX spot, FX outright, FX swap (spot-forward and forward-forward)
  - Non-Deliverable Forward (NDF)
  - NDF swap
  - Fixed Spot Roll (FSR): FSR trades have exactly the same structure as spot/forward swap trades (a spot leg and a forward leg). However, the spot rate of the FSR is specified by the customer instead of quoted by the provider. This customer-specified spot rate is then used to derive the forward points of the forward leg.

1.2.5 Batch Trades Overview

A batch trade is a collection of customer trades identified by a single portfolio ID and with the following attributes:

- A single currency pair, for example EUR/USD
- The same dealt currency, either EUR or USD
- One or more trade types: spot, outright, spot-forward swap, or forward-forward swap
- One or more value dates
- One or more accounts or funds to which the customer trades are allocated

Batch trade functionality allows the FX Grid to:

- Respond to customer requests from third-party platforms (FXall, 360T, etc.) for pre-allocated trade lists.
- Price and book pre-allocated trade list, either disclosed (one-way pricing) or undisclosed (two-way pricing) using the RFS workflow.
- Price portfolios based on netted amounts.
- Auto-cover risk.

For more information about about batch trades, see "Batch Trade Workflow" on page 18.

1.2.6 Supported Tenors

The RFS workflow supports outrights and swaps. For future dates, you can specify a broken date or a standard tenor. The FIX Client API supports the following tenors:

Table 1-1 *Supported Tenors*

Tenor	Definition
Today	Today
TOD	Today
ON	Overnight (today)
TN	Tomorrow
SP	Spot
SN	Spot next (spot+1)
*n*D	A number of days after the current business date (for example, 1D, 2D, 10D)
*n*W	A number of weeks after the current business date (for example, 1W, 2W, 3W)
*n*M	A number of months after the current business date (for example, 1M, 2M, 3M)
*n*Y	A number of years after the current business date (for example, 1Y, 2Y, 3Y)
*n*IMM	The next International Monetary Market (IMM) settlement date. IMM dates are the third Wednesday of the last month of every quarter (March, June, September, December). Entering `IMM` results in the next IMM date on or after the spot date. Entering `2IMM` results in two IMM dates after the spot date.

***Table 1-1** Supported Tenors (continued)*

Tenor	Definition
S*n*IMM	(spot + IMM) for swaps
T*n*IMM	(tomorrow + IMM) for swaps

1.2.7 Orders

The following sections describe how the FIX Client API supports order workflows:

Supported Order Types

The FIX Client API supports orders with the following `OrdType` (#40) values:

***Table 1-1** Order Types*

Order Type	OrdType (#40) Value	Description	Trading Workflow
Previously Quoted	D	The client sends new orders with a reference to a previously received executable price in `QuoteEntryID` (#299) from a quote ("Market Data Snapshot/Full Refresh" on page 75).	■ ESP ("Executable Streaming Prices (ESP) Workflow" on page 12) ■ RFS ("Request for Stream (RFS) Workflow" on page 17)
Limit	2	Orders executed according to your specifications ("Order Execution" on page 30) at the limit price or better ("Price" on page 93) until they are filled, canceled, or expired ("Order Expiry" on page 31).	Order ("Order Workflow" on page 13)

***Table 1-1** Order Types (continued)*

Order Type	OrdType (#40) Value	Description	Trading Workflow
Market	1	Orders are executed immediately at the best available price in the system. The `PegOffsetValue` (#211) field of the order must not be specified. See "PegOffsetValue" on page 97.	Order ("Order Workflow" on page 13)
Market Range	1	Orders are executed immediately at the best available price in the system as long as the slippage is within the range specified by the `PegOffsetValue` (#211) field of the order. See "PegOffsetValue" on page 97.	Order ("Order Workflow" on page 13)
Stop	3	Orders are active but do not execute until the market price reaches the order's trigger price ("StopPx" on page 94). Orders are then executed as market or market range orders depending on whether or not the `PegOffsetValue` (#211) field is specified. See "PegOffsetValue" on page 97.	Order ("Order Workflow" on page 13)
Stop Limit	4	Orders are active but do not execute until the market price reaches the order's trigger price ("StopPx" on page 94). Orders are then executed as limit orders at the order limit price or better ("Price" on page 93).	Order ("Order Workflow" on page 13)
One Cancels the Other (OCO)	N/A	OCO orders consist of two orders submitted separately and tied by their order IDs.	Order ("Order Workflow" on page 13)

If you submit an order for an unsupported order type, FX Grid responds with an Order Execution Report message with the `OrdStatus` (#39) field value of 8 (Rejected) to indicate submission failure. See "Execution Report" on page 121 for details.

Order Cancel/Replace and Amendment

If market conditions or your business needs change, you may want to modify an existing order. Typically, this involves the Order Cancel/Replace workflow (cancelling the original order and submitting a replacement order).

When multiple orders have the same rate, orders that are submitted first to the order queue are filled first. with the replacement order falling to the bottom of the queue. The FIX Client API also provides the Order Amendment workflow that allows you to amend an unfilled order's size and keep its place in the order queue.

> **NOTE:** Your organization must be configured to enable the Order Amendment workflow. Contact your Integral Solutions Manager about amending orders.

With the Order Cancel/Replace workflow, you must cancel the original order and submit a replacement order with a new client order ID (`ClOrdID` (#11)) using the Order Cancel/Replace Request message and workflow. The system assigns a new order ID (`OrderID` (#37)) to the replacement order and it joins the back of the queue. If the original order is partially filled, then the request is rejected with an Order Cancel Reject Message with the `CxlRejReason` (#102) field set to 0 (zero) "Too Late To Cancel".

With the Order Amendment workflow enabled, you use the same Order Cancel/Replace Request message and provide a new client order ID (`ClOrdID` (#11)). If the original order is unfilled and if you only change the order size, the replacement order keeps the same order ID (`OrderID` (#37)) and keeps its place in the queue. However, if the original order is partially filled or if you change the order rate, then the replacement order gets a new order ID (`OrderID` (#37)) and joins the end of the queue.

***Table 1-2** Order Cancel/Replace Compared to Order Amendment*

Comparison	Order Cancel/Replace Workflow	Order Amendment Workflow
Availability	Default behavior of the system	Must be configured. Contact your Integral Solutions Manager.
Request message sent	Order Cancel/Replace Request (see "Order Cancel/Replace Request" on page 106)	Same as cancel/replace.
Message workflow	"Order Amend and Cancel/Replace Workflow" on page 16	
Client order ID (`ClOrdID` (#11)) on request	New	Same as cancel/replace.
Request change to size only (see "Amending Orders by Size" on page 29)	Replacement order gets new order ID (loses place in queue)	Replacement order gets old order ID (keeps place in queue)
Request change to rate (see "Amending Orders by Rate" on page 30)	Replacement order gets new order ID (loses place in queue)	Same as cancel/replace.
Eligible order types	All (stops must not be triggered)	Same as cancel/replace.
Eligible order states	■ New ■ Stopped: If the order is partially filled, the request to cancel/replace is rejected with `CxlRejReason` (#102) =0 (zero) "Too Late To Cancel".	■ New ■ Partial fill: The replacement order gets a new order ID and joins the end of the queue. ■ Stopped: If the order is unfilled and you change only the order size, the replacement order gets the old order ID and retains its place in the queue. If the order is partially filled or if you change the rate, the replacement order gets a new order ID and joins the end of the queue.
Original order is partially filled	Request to cancel/replace rejected with `CxlRejReason` (#102)=0 (zero) "Too Late To Cancel"	Replacement order gets new order ID (loses place in queue)

Amending Orders by Size

Amending an order by size preserves its place in the order queue. The date/time and order ID (`ClOrdID` (#11)) of the replacement order is the same as the original order. When you amend an

order by amount, all parameters on the Order Cancel/Replace message, except the amount, must be the same as the original order.

Amending Orders by Rate

An order amended by rate loses its place in the queue. The date/time of the replacement order is updated to the date/time of the amend request and the replacement order gets a new order ID (`ClOrdID` (#11)). The amend message must match the original order in all parameters except the rate.

Order Execution

The `ExecInst` (#18) field on the New Order – Single message determines how and when the order is executed. As applicable to the order type, you can specify more than one value by separating each value with a space. For example, a stop limit order with crossing allowed ("B"), a bid trigger rate ("9"), and best-price execution ("P") would have an `ExecInst` (#18) field value of "B 9 P". For information about the field in the message, see "ExecInst" on page 92.

Table 1-3 *Execution Instructions and Applicable Order Types*

ExecInst (#18) Field Value	Description	Applicable Order Types
Empty	If the `ExecInst` (#18) field is left empty, the order is filled only by hits from other market participants at the best price that fills the order. For your order to be executable, it must be a displayed order. See "Order Visibility" on page 33 for more information.	■ Limit
B	Cross with incoming rates and with other market participants (displayed order) depending on the visibility of the order. See "Order Visibility" on page 33 for more information. Market range orders (`PegOffsetValue` (#211) specifying slippage) can be submitted as displayed orders. "Pure" market orders (`PegOffsetValue` (#211) undefined) cannot be submitted as displayed orders because they do not include a price.	■ Limit ■ Market Range ■ Stop ■ Stop Limit
B not specified	Do not cross with incoming rates. Fill only by hits from other market participants. For your order to be executable, it must be a displayed order. See "Order Visibility" on page 33 for more information.	■ Limit ■ Stop ■ Stop Limit
ST	Algo order. Your order is executed according to the parameters that you specify. See "Algo Orders" on page 38. If the algo order is a pegged order (`AlgoParameters` (#7560)=POT), the only additional execution instructions that are valid are "B" and "P". See "POT" on page 101. Other execution instructions are valid for algo orders other than pegged orders.	■ Limit ■ Market
P	Best price/market peg: Execute at the best price only (top of book)	■ Limit ■ Market ■ Stop ■ Stop Limit
P not specified	Price at depth: The order is filled with the best price in the size that allows the entire order to be filled. If "P" (best price) is not specified, then execution defaults to price at depth.	■ Limit ■ Stop ■ Stop Limit

Table 1-3 *Execution Instructions and Applicable Order Types (continued)*

ExecInst (#18) Field Value	Description	Applicable Order Types
W	VWAP (Volume Weighted Average Price): The order is filled with prices so that average execution price is equal to or better than the limit price. Some fills can be at a price worse than the order rate, but the average price remains equal to or better. For market range orders, the average execution price is kept within the market range.	■ Limit ■ Stop ■ Stop Limit
9	Bid trigger: Indicates that the bid rate is the trigger rate compared to the stop price (see "StopPx" on page 94).	■ Stop ■ Stop Limit
0	Offer trigger: Indicates that the offer rate is the trigger rate compared to the stop price (see "StopPx" on page 94).	■ Stop ■ Stop Limit
M	Mid trigger: Indicates that the mid rate is the trigger rate compared to the stop price (see "StopPx" on page 94).	■ Stop ■ Stop Limit
R	At rate: If specified, the stop is triggered if the market price equals or is less than the stop price for a SELL STOP and equals or is greater than the stop price for a BUY STOP. If not specified, the stop is only triggered if the market price is lesser/greater than the stop price (see "StopPx" on page 94).	■ Stop ■ Stop Limit
Stop trigger not specified	If a stop order is submitted with no specified stop trigger (the `ExecInst` (#18) field does not include 9, 0, M, or R), then the following is assumed: ■ A buy stop order has an offer trigger. ■ A sell stop order has a bid trigger.	■ Stop ■ Stop Limit

Order Expiry

The FIX Client API supports orders with the following `TimeInForce` (#59) values:

Expiry Type	Description	TimeInForce (#59)	Applicable Order Types
Day	The order remains active for the entire day until it is completely filled, canceled by the customer, or until the end of the business day (the roll time).	0=Day	■ Market ■ Limit ■ Stop ■ Stop Limit
Good Till Cancel (GTC)	The order remains active until either it is completely filled or is canceled by the customer.	1=GTC	■ Market ■ Limit ■ Stop ■ Stop Limit

Expiry Type	Description	TimeInForce (#59)	Applicable Order Types
Immediate or Cancel (IOC)	The order is matched with the available order book in the system after the submission. For market and limit orders, any unfilled amount is immediately canceled with an Execution Report message with `OrdStatus` (#39)=4 and `ExecType` (#150)=4 (Canceled). Previously quoted orders do not receive cancellation messages for partial fills. If `MinQty` (#110) is specified, the first fill must be greater than or equal to the minimum quantity and lesser than or equal to the order quantity. Subsequent fills, if any, have no size requirement. If `MinQty` (#110) is not specified, the first fill may of any size. Subsequent fills, if any, have no size requirement. Not specifying a minimum quantity for IOC orders may result in small, random order fills. This expiry type is not valid for component orders of an OCO order or for stop/stop limit orders. Orders of these types with this expiry type fail submission.	3=IOC	■ Market ■ Limit ■ Previously Quoted
Fill or Kill (FOK)	The order is matched with the available order book in the system after the submission. The order amount is either completely filled with one or more fills or canceled in its entirety. No partial fill is allowed. The first fill may be of any size. Subsequent fills may be of any size and must completely fill the order or the entire order is cancelled. The unfilled order is canceled with an Execution Report message with `OrdStatus` (#39)=4 and `ExecType` (#150)=4 (Canceled). This expiry type is not valid for component orders of an OCO order or for stop/stop limit orders. Orders of these types with this expiry type fail submission.	4=FOK	■ Market ■ Limit
Good Till Date/Time (GTD)	The order remains active until one of following conditions is met: ■ Fully filled ■ Canceled/Replaced by the customer ■ Expired For GTD orders, the relative expiration time must be specified in the `ExpireTime` (#126) field in the format *YYYYMMDD-hh:mm:ss.sss*. The date portion of the value is ignored. The time portion specifies the amount of time before the quote request expires. For example, for an order that expires in ten seconds, the value would be 00000000-00:00:10.000. Applicable to all order types except previously quoted. See "Order Workflow" on page 13.	6=GTD	■ Market ■ Limit ■ Previously Quoted ■ Stop ■ Stop Limit

Partial Fills

NOTE: Partial fills apply only to the Order trading workflow. See "Trading Workflows" on page 12 for information about trading workflows.

You can specify how limit orders are filled with the `MinQty` (#110) field in the New Order – Single message:

- Partial fill (`MinQty` (#110) set to zero): The order amount can be filled multiple times with any size until the entire amount is filled.
- Partial fill with market minimum (`MinQty` (#110) greater than zero): The first fill must be equal to or greater than the market minimum defined by the `MinQty` (#110) field. Subsequent fills can

be of any size. If the residual unfilled order amount is less than the market minimum, then the order is considered fully filled and done. The residual amount is implicitly canceled by the server.

- No partial fill (`MinQty` (#110) equals the order size): The order amount must be filled in its entirety with exactly one fill.

If you allow partial fills of your order, the server executes against your order only with new quotes that match your order. The server may execute multiple fills against a single liquidity provider as long as your order is open and incoming quotes from the provider match your order.

For more details about partial fills, see the field "MinQty" on page 95.

Minimum Order Size

NOTE: Partial fills apply only to the Order trading workflow. See "Trading Workflows" on page 12 for information about trading workflows.

The FIX server can be configured for a minimum order size that is used at two points in the Order trading workflow:

- Limit order submission: When the order is first submitted, if the order amount is lower than the server's minimum order amount, the order submission fails.
- Partial fills: After each partial fill, if the residual amount of the order is lower than the minimum order amount, the residual amount is canceled automatically and an order cancellation message is sent to the client. The value of the `MinQty` (#110) field specifies whether the order allows partial fills and the field's value overrides the minimum order size setting on the server. For more details, see the field "Partial Fills" on page 32.

Order Visibility

You can determine the amount of the limit order that is visible to other market participants based on the value of the `MinQty` (#110), `MaxShow` (#210), and `OrderQty` (#38) fields in the New Order – Single message:

- Hidden: The order is hidden and is not visible to other customers. The order is matched with incoming rates only. The `MaxShow` (#210) value is 0 (zero) or the `MaxShow` (#210) field is not included in the message.
- Display: The order is completely visible to other customers. The full order amount is disclosed. The `MaxShow` (#210) value equals the `OrderQty` (#38) value.
- Hidden that transitions to display: If the order's minimum fill (`MinQty` (#110)) is greater than its show amount (`MaxShow` (#210)), then the order is hidden and not displayed to other market participants. The order is only visible after the minimum fill amount is achieved. Conversely, if `MinQty` (#110) is less than `MaxShow` (#210), the displayed order is always broadcast. For this behavior to apply, the order in question must allow partial fills (`MinQty` (#110) is less than `OrderQty` (#38)) and the order must not have an FOK time in force (`TimeInForce` (#59) is not 4 (FOK)).
- Iceberg: The order is visible to other customers, but only a fraction of the actual order amount is displayed. The `MaxShow` (#210) value is less than the `OrderQty` (#38) value.

NOTE: Regardless of order visibility, if you want your order crossed with incoming market prices, you must set the `ExecInst` (#18) field of the New Order – Single message to "B". For example, if you submit a hidden order (`MaxShow` (#210) is zero or not included in the message) and do not include "B" as a value of the `ExecInst` (#18) field, then your order is not crossed with incoming market prices and cannot be filled.

For more details, see:

- "OrderQty" on page 93
- "MaxShow" on page 96
- "ExecInst" on page 92

Duplicate Order IDs

The client should avoid sending a New Order – Single message with a duplicate `ClOrdID` (#11) value. Generally, if an order has been filled or partially filled, the `ClOrdID` (#11) value of the order cannot be used for a certain period of time. If the order has not been filled and has reached a terminal state (Cancelled or Expired), then the `ClOrdID` (#11) value can be reused.

The server maintains a time-based cache of `ClOrdID` (#11) values of active and completed orders for the current session. The typical time-to-live for a `ClOrdID` (#11) value in the cache is one hour. If the server receives an execution request for a duplicate `ClOrdID` (#11) before the ID has expired in the cache, the server rejects the execution request.

Furthermore, to prevent duplicate executions, the server fails the submission of New Order – Single messages with the `PossDupFlag` (#43) or `PossResend` (#97) fields set. These fields are set by FIX engines when they resend messages, such as after recovering from a network outage. Because the server does not store all past `ClOrdID` (#11) values, the server cannot determine whether the same ID was used previously and it fails the submission of any order with the `PossDupFlag` (#43) or `PossResend` (#97) field set.

One-Cancels-the-Other (OCO) Orders

The FIX Client API represents OCO orders as two or more component orders that you submit separately and that are linked by their respective order IDs. A fill or partial fill of one component order results in the system cancelling the other component orders in the chain.

The `ContingencyType` (#1385)=1 (OCO) value indicates that an order is a component of an OCO order chain. The `ClOrdLinkID` (#583) field of a component order contains the ID of the next linked order in the chain.

The overall life cycle of an OCO order chain with two orders is:

1. You submit the first component order. The order's `TimeInForce` (#59) expiry must be something other than IOC or FOK.
2. Receive an Execution Report message indicating that the first component order is valid with the `OrdStatus` (#39) field set to 0 (zero, New).
3. Submit the second component order with `ContingencyType` (#1385)=1 and the `ClOrdLinkID` (#583) set to the client order ID `ClOrdID` (#11) of the first component order.
4. The system performs the following validation:

- Ensures that the first component order is still in the NEW state (has not been expired, cancelled, or filled).
- Checks that the `TimeInForce` (#59) expiry of the second component order is not IOC or FOK.

5. If the second component order passes validation:
   - You receive an Execution Report for the second component order.
   - The system sets the `ContingencyType` (#1385) and the `ClOrdLinkID` (#583) of the first component order to indicate that it is part of an OCO order and to link it to the second component order.
6. If either one of the component orders is filled or partially filled, then the system cancels the other component order. If one component order is expired or cancelled, then the other order is not cancelled and remains in effect until it is filled, expired, or cancelled. In all other aspects (execution, expiry, status), each component order behaves the same as any other order with the same expiry and execution characteristics.

One-Updates-the-Other (OUO) Orders

The FIX Client API represents OUO orders like OCO orders: two or more component orders submitted separately and linked by their respective order IDs. However, instead of cancelling component orders, a fill or partial fill of one component order results in the system amending the size of the other orders in the order chain.

The orders in an OUO order chain can be reduced in two ways:

- `ContingencyType` (#1385)=3 (OUO - Absolute Quantity Reduction): Orders in the chain are reduced by the absolute amount. For example, if a component order is filled for 1M, all other orders in the chain are reduced by 1M.
- `ContingencyType` (#1385)=4 (OUO - Proportional Quantity Reduction): Orders in the chain are reduced by a relative amount. For example, if a component order is filled by 50% of its total size, all other orders in the chain are reduced by 50%.

The `ClOrdLinkID` (#583) field of a component order contains the ID of the linked order.

The overall life cycle of an OUO order chain with two orders is:

1. You submit the first component order. The order's `TimeInForce` (#59) expiry must be something other than IOC or FOK.
2. Receive an Execution Report message indicating that the first component order is valid with the `OrdStatus` (#39) field set to 0 (zero, New).
3. Submit the second component order with `ContingencyType` (#1385)=3 or 4 and the `ClOrdLinkID` (#583) set to the client order ID `ClOrdID` (#11) of the first component order.
4. The system performs the following validation:
   - Ensures that the first component order is still in the NEW state (has not been expired, cancelled, or filled).
   - Checks that the `TimeInForce` (#59) expiry of the second component order is not IOC or FOK.
5. If the second component order passes validation:
   - You receive an Execution Report for the second component order.

- The system sets the `ContingencyType` (#1385) and the `ClOrdLinkID` (#583) of the first component order to indicate that it is part of an OUO order chain and to link it to the second component order.

6. If either one of the component orders is filled or partially filled, the server sends an Execution Report message with `ExecType` (#150)=D (Restated) with an updated order quantity for the other orders in the OUO chain. If the unfilled amount on any of the component orders falls below the minimum trade size or the minimum fill quantity specified on the order, the order is cancelled. This order cancellation does not affect the amount on the other orders. If one component order is expired or cancelled, then the other orders in the chain are not cancelled and remains in effect until it is filled, expired, or cancelled. In all other aspects (execution, expiry, status), each component order behaves the same as any other order with the same expiry and execution characteristics.

Order Persistence

By default, the FIX server cancels all open orders submitted during a session when the FIX users logs out.

The FIX server can be configured to persist orders in the system and allow execution when the FIX user is logged out. Contact your Integral Solutions Manager to enable persistent orders.

When the FIX user logs back in, the FIX client must query the FIX server for the current status of all orders that were open when the user logged out.

With persistent orders, trades that are executed while logged out are sent via STP download, if configured.

Orders with the following characteristics are currently not persisted:

- IOC orders (`TimeInForce` (#59)=3)
- FOK orders (`TimeInForce` (#59)=4)
- Market orders (`OrdType` (#40)=1)

Orders that cannot be persisted are cancelled by the FIX server when the FIX user logs out whether or not persistent orders are enabled. Contact your Integral Solutions Manager if you require other order types to be persisted.

The FIX server maintains price subscriptions to currency pairs for any persisted orders that are still open.

When the FIX server is brought down for planned maintenance, all order execution is suspended and any open IOC, FOK, and market orders are cancelled.

Order Status

NOTE: Partial fills apply only to the Order trading workflow. See "Trading Workflows" on page 12 for information about trading workflows.

The server maintains an order's status as it transitions from one state to another as a result of business and system events. The server notifies the FIX client of order status with the `OrdStatus` (#39) field of the Order Execution Report message. See "Execution Report" on page 121 for details.

The following diagram illustrates an order's state transitions and events. Circles represent order status as reported by the `OrdStatus` (#39) field. The lines between status circles represent state transitions are labelled with the effecting event names as represented by the `ExecType` (#150) field of the Order Execution Report message.

Note that the same event can result in different orders statuses. The destination status is determined by other attributes of the order. For example, an order with status "New" can transition to "Partially filled" or to "Filled" with a trade event depending on whether the order allows partial fills and whether or not the trade completely fills the order.

The server may not send a message for the "New" (`ExecType` (#150)=A) order state. If the order can be filled immediately, the server may skip the "New" ack message and send only a "Trade" Execution Report message (`ExecType` (#150)=F) to indicate a fill. If your organization's workflow requires order acknowledgement with the "New" Execution Report message (`ExecType` (#150)=0), please contact your Integral Solutions Manager.

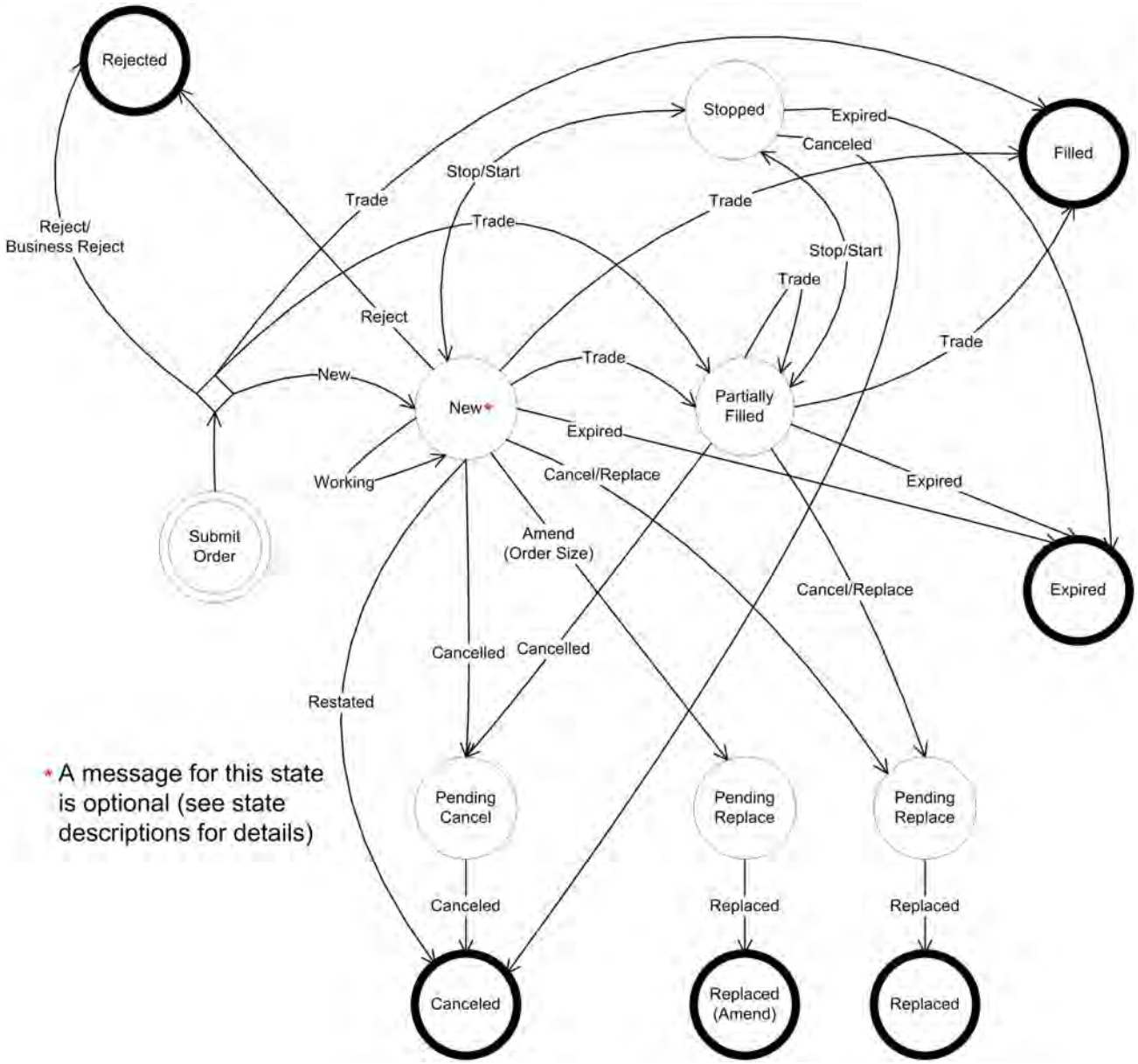

Figure 1-1 *Order State Transitions*

Table 1-4 *Order Status*

Order Status, value of OrdStatus (#39) field	Description
A=Pending New	An order has been received by server. Only applicable to the RFS workflow (see "Request for Stream (RFS) Workflow" on page 17.
0 (zero)=New	The server has confirmed that the order is valid. The order has been successfully entered into the server's order management system. This status may be skipped. This status applies to stop and stop limit orders when they have been triggered and are working. The server may not send a message for the "New" (ExecType (#150)=A) order state. If the order can be filled immediately, the server may skip the "New" ack message and send only a "Trade" Execution Report message (ExecType (#150)=F) to indicate a fill. If your organization's workflow requires order acknowledgement with the "New" Execution Report message (ExecType (#150)=0), please contact your Integral Solutions Manager.
1=Partial fill	The order has been partially filled on the server with a residual amount unfilled.
2=Fill	The order has been completely filled.
8=Rejected	The server failed to confirm the validity of the order and the order submission has failed.
7=Stopped	The server failed to confirm the validity of the order and the order submission has failed.
6=PendingCancel	A cancel request has been received by the server and is being processed.
4=Canceled	An order and any residual amount of the order has been canceled by one of the following mechanisms: ■ Customer action (for example, canceling the order in FX Grid) ■ Unfilled FOK orders ■ Unfilled or partially filled IOC ■ As a result of a fill of an associated component order of an OCO order ■ Unsolicited by the server (for example, a stop order submitted with an invalid time in force)
E=PendingReplace	A cancel/replace request has been received by the server and is being processed.
5=Replaced	An existing order has been replaced by a new order or amended.
C=Expired	The order has expired. Any residual amount of the order is canceled.

1.2.8 Algo Orders

Algo orders allow you to execute your order over time, minimizing your order's effect on the market so that you can fulfill your liquidity needs at a price closer to the true market price.

To indicate that an order is an algo order, you include the value "ST" in the ExecInst (#18) field. See "Order Execution" on page 30.

In addition to standard orders parameters, such as currency pair, order size, and order type, algo orders involve parameters that you use to specify how your order is divided into slices, priced, and

executed over time. These additional parameters allow you to control the following aspects of your algo order:

- Time: the total duration of the algo, the interval between order slices (fixed time or randomized), and how long the order slice remains passive (see "Time (Algo Start and Duration)" on page 39 and "Time (Order Slice Characteristics)" on page 39)
- Size: the size of individual order slices (fixed amount or randomized, see "Slice Size" on page 40)
- Price: the price of individual order slices (limit, market, or pegged to a market price, see "Slice Price" on page 40)
- Action: how order slices are executed and how any unfilled amount is handled at the end of the order's duration (see "Action" on page 41)

Time (Algo Start and Duration)

Table 1-1 *Time Attributes (Algo Start and Duration)*

Algo Attribute	FIX Field	Description
Begin time (absolute)	`EffectiveTime` (#168)	The absolute time in GMT when the algo begins executing (for example, "20120324-14:00:00" for 14:00 in the afternoon of March 24, 2012)
End time (absolute)	`ExecEndTime` (#7556)	The absolute time in GMT when the algo stops executing
Begin time (relative)	`ExecEffPeriod` (#7564)	The relative time when the algo begins executing (for example, "01:00:00" for 1 hour after order submission)
End time (relative)	`ExecEndPeriod` (#7565)	The relative time when the algo stops executing

Time (Order Slice Characteristics)

Table 1-2 *Time Attributes (Order Slice Characteristics)*

Algo Attribute	Attribute Name	FIX Field	Description
Peg time for each slice	PegTime	`AlgoParameters` (#7560)= `PT`	The time during which each order slice is broadcast to the market but remains passive and is not matched with market prices. Other market participants may hit the slice while it is passive. The slice begins matching after it is hit or after the peg time expires, whichever comes first. A peg time of zero indicates that the slice is pegged for the entire duration.
Interval between slices	SlcIntrvl	`AlgoParameters` (#7560)= `SI`	The interval between slices. If the interval is fixed, this value is the interval duration. If the interval is randomized, this value is the upper boundary of random values.

Table 1-2 *Time Attributes (Order Slice Characteristics) (continued)*

Algo Attribute	Attribute Name	FIX Field	Description
Random intervals	SlcIntrvlRndmizr	`AlgoParameters` (#7560)= `SIR`	Whether or not the interval between slices is randomized. The interval is a random period between the slice interval (`SI`, SlcIntrvl) and the minimum slice interval (`SMI`, SlcMinIntrvl)
Minimum interval	SlcMinIntrvl	`AlgoParameters` (#7560)= `SMI`	The minimum interval between slices if the intervals are randomized. Slice intervals are randomized between this value and the value of the slice interval value (`SI`, SlcIntrvl).

Slice Size

Table 1-3 *Slice Size Attributes*

Algo Attribute	Attribute Name	FIX Field	Description
Minimum regular size	SlcRegularSize	`AlgoParameters` (#7560)= `SRS` ()	If slice size is randomized, this value is used in two ways. First, it is the low end of the random size range. Second, random sizes are rounded to a multiple of this value.
Fixed size	SlcSize	`AlgoParameters` (#7560)= `SS` ()	The fixed size of each slice
Random size	SlcSizeRndmzr	`AlgoParameters` (#7560)= `SSR` ()	Whether or not the slice size is randomized. The size is a random value between the slice size (`SS`, SlcSize) and the regular slice size (`SRS`, SlcRegularSize) or, if the top-of-book percent (`TOBP`, TOBPercent) is specified, between the calculated top-of-book amount and the slice size (`SS`, SlcSize).
Size based on top of book	TOBPercent	`AlgoParameters` (#7560)= `TOBP` ()	When the slice size is not randomized, this value is used to calculate the size as a percentage of the top-of-book size. When the slice size is randomized, then the size is a random value between the regular slice size (`SRS`, SlcRegularSize) and the calculated percentage of the top-of-book size. Note that it is best to set a regular size to avoid hitting a small size at the top of the book and to avoid irregular fills.

Slice Price

If your algo is submitted as a limit order or a market order, the order slices are executed as specified by the order. You can also peg your algo's price to the market using the following parameters:

Table 1-4 *Slice Price Attributes*

Algo Attribute	Attribute Name	FIX Field	Description
Offset type	PegOffsetType	`AlgoParameters` (#7560)= `POT`	■ Primary: Peg to bid if buying, offer if selling ■ Market: Peg to bid if selling, offer if buying ■ Mid: Peg to midpoint between the bid and the offer ■ Transition: First pegged to primary, then over the offset increment interval (`PII`, PegOffsetIncrIntrvl), improves the price (increase the bid, decrease the offer in increments by the offset increment (`POI`, PegOffsetIncr), and stops incrementing when the offset equals zero.
Offset value	PegOffset	`AlgoParameters` (#7560)= `PO`	The amount in pips that is added to the price of the peg. Can be positive or negative.
Offset increment	PegOffsetIncr	`AlgoParameters` (#7560)= `POI`	The fixed amount in pips used to adjust the price offset for transition pegged orders
Offset increment interval	PegOffsetIncrIntrvl	`AlgoParameters` (#7560)= `PII`	The interval at which the offset increment is applied for transition pegged orders. Specified in milliseconds.
Random offset increment	PegOffsetIncrRndmzr	`AlgoParameters` (#7560)= `PIR`	Whether or not the offset increment is randomized for transition pegged orders. When the offset increment is randomized, the increment value is a random value between zero and the offset increment (`POI`, PegOffsetIncr).

Action

Table 1-5 *Action Attributes*

Algo Attribute	Attribute Name	FIX Field	Description
Slice execution	SlcFillOrKill	`AlgoParameters` (#7560)= `SFOK`	Slices should be filled completely with a single fill or not at all. Each slice is executed as FOK with no partial fills allowed. This parameter is used only if the slice size (`SS`, SlcSize) is fixed.
Action at expiration	ActnOrdrExpr	`AlgoParameters` (#7560)= `AE`	Action taken at the end of the algo's duration if there is an unfilled amount remaining. If not specified, the order is cancelled at expiration. 1=Fill @ Market 2=Cancel Order

Example 1-1 *Underlying order: Sell 70M EUR/USD market order with GTC time in force*

Algo parameters:
`168=…18=ST…7560=SS~0.27 SI~00:01:50 SIR~Y SMI~00:01:00 AE~2…7565=00:20:00…`

- Begin time = Now (`EffectiveTime` (#168) not defined)
- Execution type = Algo (`ExecType` (#18)=`ST`)
- Slice size = 27% of top of book (`SS~0.27`)
- Slice interval = 1 minute 50 seconds (`SI~00:01:50`)
- Slice interval randomized = Yes (`SIR~Y`)
- Slice minimum interval = 1 minute (`SMI~00:01:00`)
- Action at expiration = Cancel order with any remaining unfilled amount (`AE~2`)
- End time = 20 minutes (`ExecEndPeriod` (#7565)=`00:20:00`)

Summary: Sell 70M EUR vs. USD at the market price with an order slice size that is 27% of the top of book size and with a random interval between slices ranging from 1 minute to 1 minute 50 seconds.

1.2.9 Integration with an Order Management System (OMS)

To integrate your in-house OMS with the execution services provided by the FX Grid, you use the FIX Client API to submit your orders into a staging area where they are picked up and batched for execution.

Supported Messages

OMS integration workflow involves the following message types:

- "New Order – Single" on page 90
- "Execution Report" on page 121
- "Order Status Request" on page 119
- "Order Cancel Request" on page 105
- "Order Cancel Reject" on page 115

New Order – Single

The client sends a New Order – Single message to submit an order in FX Grid.

The order can be an execution request on a streaming price (`OrdType` (#40)=D) or a limit order (`OrdType` (#40)=2) that is crossed with quotes on the server and broadcast to other market participants. See "Trading Workflows" on page 12 for more information about trading workflows.

Table 1-1 *OMS-specific New Order – Single Message Values*

Tag	Field Name	Req'd	Value	FIX Format	Description
21	HandlInst	N	1	char	3=Manual order, best execution

***Table 1-1** OMS-specific New Order – Single Message Values (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
44	Price	See descr.	—	Price	FX spot: The order limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. FX swap: Not applicable.
59	TimeInForce	Y	■ 1=GTC	char	Other times in force are not supported for OMS integration. See "Order Expiry" on page 31 for a description of order expiry types.
64	FutSettDate	N	—	LocalMktDate	For OMS integration, this date must be a future date or the order is rejected.
193	FutSettDate2	N	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	For OMS integration, this date must be a future date or the order is rejected.

1.2.10 Server Synchronization

Your servers initiating the FIX connection to FX Grid should be synchronized with the NTP pool. Unsynchronized servers result in a `RequestValidationError.TooLateToEnter` error. For more information about using the NTP pool, please use the following link:
http://www.pool.ntp.org/en/use.html

1.2.11 Business Day End and Start

You must configure your FIX engine for an end of day at 17:00:00 EST and a start of day at 17:10:00 EST. Note that the EST time zone is subject to daylight saving time. Consult your FIX engine's documentation to find out how to set business day end and start.

1.2.12 Sequence Number Reset

Resetting the sequence number when the client logs on is optional (`ResetSeqNumFlag` (#141)=Y on the Logon message).

You must configure your FIX engine to reset the sequence number `MsgSeqNum` (#34)=0 at the end of day and not on disconnect or logout. The end of the business day is 17:00:00 EST irrespective of daylight saving time.

To reset sequence numbers once a day after the business day end, the following session-level FIX parameters need to be set on the server and the client with the following values:

***Table 1-1** Server-side FIX Sequence Number Parameters*

Parameter	Value
`ResetOnDisconnect`	N
`ResetOnLogout`	N
`StartTime`	17:00:05

Table 1-1 *Server-side FIX Sequence Number Parameters (continued)*

Parameter	Value
EndTime	17:00:00

Table 1-2 *Client-side FIX Sequence Number Parameters*

Parameter	Value
ResetOnDisconnect	N
ResetOnLogout	N
StartTime	17:00:10
EndTime	16:59:00

1.2.13 Event Sequencing

The FIX Client API handles messages and trading events on a first-come first-served basis.

1.2.14 Client Roles

When trading through FX Grid as a FIX client, you are either a *direct customer* or a *facilitator*:

- Direct customer: You trade directly with any number of liquidity providers in a one-to-one or one-to-many basis. The prices you see are a market data stream composed of rates from *n* providers, where *n* equals one to the number of providers for which you provisioned to trade. Individual trades are done between you and a single provider.
- Facilitator: You are a sales dealer or facilitator organization who initiates trading with any number of liquidity providers in a one-to-one or one-to-many basis on behalf of a customer. The prices you see are a market data stream composed of rates from *n* providers, where *n* equals one to the number of providers for which you are provisioned to trade. Individual trades are done between the customer and a single provider.

Your client role determines the values of the additional message-routing fields in the message header:

- `OnBehalfOfCompID` (#115)
- `OnBehalfOfSubID` (#116)
- `DeliverToCompID` (#128)

See "Organization and User Identification" on page 54 for details descriptions of the fields and "Summaries of ID Values" on page 58 for a summary of values for both client roles and all message types.

1.2.15 Quote Types

The FIX Client API distinguishes between two types of quotes, each with their own distinct representation and workflow. Each Market Data Snapshot/Full Refresh message received from FX Grid contains only one type of quote.

Multi-price Quotes

For multi-price quotes, each price in a message can be treated like a separate limit order and can be dealt upon independently. You can send an order for each quote entry independently of the others and you can send multiple orders before you receive a refresh message.

Multi-price quotes are represented in the FIX message as repeating groups of related fields. The value of the `MDEntryPositionNo` (#290) field in each group is zero, indicating that the prices are parts of a multi-price quote. See "Market Data Snapshot/Full Refresh" on page 75 for information about quote entry representation.

Multi-tier Quotes

For multi-tier quotes, each price in a message is considered a tier. There is one bid and one offer rate at each tier. The size for the bid and offer rates may differ.

The tiers are mutually exclusive. You can deal on only one tier of a quote at a time.

Also, the size of the order you place for a tier must be less than or equal to the size of the tier and greater than the size of the previous tier. For example, if the prices are tiers of 10M, 20M and 30M, the order you place for 25M must be submitted for the 30M tier.

Like multi-price quotes, multi-tier quotes are represented in the FIX message as repeating groups of related fields. If the value of the `MDEntryPositionNo` (#290) field in the repeating group is greater than zero, it indicates both a multi-tier quote and the price's tier position. For example, for a quote with three tiers, the value of the `MDEntryPositionNo` (#290) field in each tier is "1", "2", and "3" respectively.

See "Market Data Snapshot/Full Refresh" on page 75 and for information about multi-tier quote representation.

1.2.16 Provider Names with Merged Prices and Provider Priority

The system can be configured to include liquidity provider names in the `MDEntryOriginator` (#282) field in merged prices when prices are sent in a single message as an aggregated book. When provider names are not included, a merged price is labeled with the name of the organization that requested the price.

How provider names are shown for aggregated books and merged prices depends on how the system is configured, how the price tiers are defined, and provider priority.

You must work with your Integral Solutions Manager to enable this feature.

Provider priority for the following examples is:

1. A_Bank
2. B_Bank

3. C_Bank
4. D_Bank
5. E_Bank
6. F_Bank

For all examples, the aggregate book with unmerged prices and no provider priority applied is:

Table 1-1 *Aggregate Book Source Data*

LP Name	Size	Bid	Offer	Size	LP Name
A_Bank	2,000,000	1.28919	1.28928	1,000,000	D_Bank
B_Bank	1,000,000	1.28917	1.28929	1,000,000	B_Bank
D_Bank	1,000,000	1.28915	1.28930	2,000,000	C_Bank
C_Bank	2,000,000	1.28911	1.28931	2,000,000	A_Bank
E_Bank	1,000,000	1.28898	1.28934	1,000,000	E_Bank
B_Bank	3,000,000	1.28898	1.28939	3,000,000	B_Bank
F_Bank	1,000,000	1.27329	1.29024	1,000,000	F_Bank

Full Book (Unmerged Prices)

The provider associated with each price in the book is sent in the `MDEntryOriginator` (#282) field of the price. If more than one provider has the same price, then the providers appear in the book in priority order.

In the following example, the quote from B_Bank is higher in the aggregated book than E_Bank because of its higher priority.

Table 1-2 *Full Book with Unmerged Prices*

LP Name	Size	Bid	Offer	Size	LP Name
A_Bank	2,000,000	1.28919	1.28928	1,000,000	D_Bank
B_Bank	1,000,000	1.28917	1.28929	1,000,000	B_Bank
D_Bank	1,000,000	1.28915	1.28930	2,000,000	C_Bank
C_Bank	2,000,000	1.28911	1.28931	2,000,000	A_Bank
B_Bank	**3,000,000**	**1.28898**	1.28934	1,000,000	E_Bank
E_Bank	1,000,000	1.28898	1.28939	3,000,000	B_Bank
F_Bank	1,000,000	1.27329	1.29024	1,000,000	F_Bank

FIX settings in Market Data Request message for the aggregated full book with unmerged prices:

- `DeliverToCompID` (#128)=ALL or comma-separated list of providers to receive aggregate book in a single market data message
- `MarketDepth` (#264)=0 (Full Book)
- `AggregatedBook` (#266)=N (Multiple entries per side per price allowed)

- `AggregationType` (#7548)=Not applicable

Full Book (Merged Prices)

The provider associated with each price in the aggregated book is sent in the `MDEntryOriginator` (#282) field of the price. If more than one provider has the same price, then all of the prices are merged into a single price entry in the book and the highest priority provider is sent with that price with a plus symbol (+) to indicate the merged price.

In the following example, the bid prices of 1.28898 have been merged and assigned B_Bank as the provider name because B_Bank has a higher provider priority than E_Bank.

Table 1-3 *Full Book with Merged Prices*

Pseudo Name	Size	Bid	Offer	Size	Pseudo Name
A_Bank	2,000,000	1.28919	1.28928	1,000,000	D_Bank
B_Bank	1,000,000	1.28917	1.28929	1,000,000	B_Bank
D_Bank	1,000,000	1.28915	1.28930	2,000,000	C_Bank
C_Bank	2,000,000	1.28911	1.28931	2,000,000	A_Bank
B_Bank+	**4,000,000**	**1.28898**	1.28934	1,000,000	E_Bank
F_Bank	1,000,000	1.27329	1.28939	3,000,000	B_Bank
			1.29024	1,000,000	F_Bank

FIX settings in Market Data Request message for the aggregated full book with merged prices:

- `DeliverToCompID` (#128)=ALL or comma-separated list of providers to receive book in a single market data message
- `MarketDepth` (#264)=0 (Full Book)
- `AggregatedBook` (#266)=Y (One book entry per side per price)
- `AggregationType` (#7548)=F (Full Book)

Multi-tier Book (Merged Prices)

Each price in a multi-tier book is considered a tier. There is one bid and one offer price at each tier. See

The provider associated with the top tier is used to determine the name used for all tiers of the book. All prices in the book have the same name in the in the `MDEntryOriginator` (#282) field as the provider with the best bid/offer price.

If the tiers are requested as 1M, 2M, 3M, 5M,and 10M, then the aggregated book is:

Table 1-4 *Multi-tier Prices*

Pseudo Name	Size	Bid	Offer	Size	Pseudo Name
A_Bank	1,000,000	1.28919	1.28928	1,000,000	D_Bank
A_Bank	2,000,000	1.28919	1.28929	2,000,000	D_Bank

***Table 1-4** Multi-tier Prices (continued)*

Pseudo Name	Size	Bid	Offer	Size	Pseudo Name
A_Bank	3,000,000	1.28917	1.28930	3,000,000	D_Bank
A_Bank	5,000,000	1.28911	1.28931	5,000,000	D_Bank
A_Bank	10,000,000	1.28898	1.28939	10,000,000	D_Bank

All bid prices are assigned the A_Bank LP name because A_Bank has the best bid price.

All offer prices are assigned the D_Bank LP name because D_Bank has the best offer price.

FIX settings in Market Data Request message for multi-tier aggregated book with merged prices:

- `DeliverToCompID` (#128)=ALL or comma-separated list of providers to receive book in single market data message
- `MarketDepth` (#264)=5 (5 tiers)
- `RequestedSize` (#7546)=1M,2M,3M,5M,10M
- `AggregatedBook` (#266)=Y (One book entry per side per price)
- `AggregationType` (#7548)=F (Full Book)

1.2.17 USIs and Mid Mark Data

Unique Swap Identifiers (USIs) and mid-mark data are essential information for trading under SEF regulations. The FIX Client API handles this information on the following messages:

***Table 1-1** USI and Mid-Mark Data by Message and Source*

Message Type	Data on Message	
	Outgoing from client	**Incoming from FX Inside**
Quote Request (`MsgType` (#35)=R)	USI, USI Prefix	
Quote (`MsgType` (#35)=S)		Mid-rate
New Order – Single (`MsgType` (#35)=D)	USI, USI Prefix, Mid-rate	
Execution Report (`MsgType` (#35)=8)		USI, USI Prefix, Mid-rate
Trade Capture Report (`MsgType` (#35)=AE)		USI, USI Prefix, Mid-rate

USI Fields

The FIX Client API holds USI information in the following fields:

Tag	Field Name	Req'd	Value	FIX Format	Description
9374	USINamespace	N	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

Tag	Field Name	Req'd	Value	FIX Format	Description
9375	USIFarNamespace	N	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9376	USI	N	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

NOTE: USI fields are not included in messages by default. Your organization must be configured to handle USI fields.

Client-generated USI

If the client organization is configured to generate the USI, the USI client sends the USI on the Quote Request message or New Order – Single message. The last USI received for a request ID or order ID is stored by FX Inside and sent back to the client in Execution Report and Trade Capture Report messages.

Integral-generated USI

If FX Inside generates the USI, FX Inside sends the USI to the client in Execution Report and Trade Capture Report messages.

USIs and Transaction IDs

Per the CFTC specification, a USI is a combination of a ten-character namespace and a transaction ID. USIs are handled in the following manner:

- If a USI namespace is specified in a Quote Request or New Order – Single message in `USINamespace` (#9374) or `USIFarNamespace` (#9375), the USI is checked to see if it begins with the USI namespace. If so, the transaction ID is derived by removing USI namespace from the USI.
- If a USI namespace is specified and the USI does not begin with the namespace, the USI is treated as transaction ID and final USI is the USI namespace plus the transaction ID.
- If a USI namespace is not specified, the transaction ID and USI namespace are considered as empty.

The USI namespace and final USI as derived are sent to the client in Execution Report messages and Trade Capture Report messages.

Mid Mark

The FIX Client API holds mid-mark information in the following fields:

Tag	Field Name	Req'd	Value	FIX Format	Description
631	MidPx	N	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

Customer-generated Mid Mark

The mid-rate information that the client sends on the New Order – Single message is stored and used in subsequent Execution Report and Trade Capture Report messages in the workflow.

Integral-generated Mid Mark

If the client does not send mid-rate information on the New Order – Single message, the mid rate is taken from the accepted Quote message that is used to match with the originating of New Order – Single message and used in subsequent Execution Report and Trade Capture Report messages in the workflow. The mid-rate fields are not sent by default.

For swap trades, if a provider sends only mid swap points, then FX Inside sends the mid swap points in the far mid-rate field (`MidPxFl` (#7631) and sets the near mid-rate field (`MidPx` (#631) to empty or zero.

1.3 FIX Implementation

The interface defined by the FIX Client API conforms to the FIX 4.3 specifications with specific extensions using the FIX 4.4 protocol. The FIX Protocol Organization provides a complete reference to the protocol at:

http://www.fixprotocol.org

1.3.1 Message Examples

NOTE: The FIX Protocol uses the nonprinting ASCII code 1 character (also known as SOH, Start of Header, 0x01, and Ctrl-A) as a field/value delimiter. This character has been replaced with the pipe character "|" to aid readability.

Message examples are packaged with this guide in a Word document.

1.3.2 Supported Message Types

The FX Inside FIX Client API supports the following FIX messages types:

Table 1-1 *Supported Message Types*

Message Type—MsgType (#35) Field Value	Message Name	Inbound to Client (I), Outbound from Client (O), or Bidirectional (B)
Session Management		
A	Logon	B
5	Logout	B
0 (zero)	Heartbeat	B
1	Test Request	B
3	Session-Level Reject	B
Market Data and Quote Workflow		
V	Market Data Request	O
W	Market Data Snapshot/Full Refresh	I
Y	Market Data Request Reject	I
R	Quote Request	O
AG	Quote Request Reject	I
S	Quote	I
Z	Quote Cancel	B
Trading Workflow		
D	New Order - Single	O
AB	New Order – Multileg	O
F	Order Cancel Request	O
G	Order Cancel/Replace Request	O
9	Order Cancel Reject	I
q	Order Mass Cancel Request	O
r	Order Mass Cancel Report	I
H	Order Status Request	O
8	Order Execution Report	I
Position Management		
AN	Request for Positions	O
AO	Request for Positions Ack	I
AP	Positions Report	I

Table 1-1 *Supported Message Types (continued)*

Message Type—MsgType (#35) Field Value	Message Name	Inbound to Client (I), Outbound from Client (O), or Bidirectional (B)
Post-trade Messages		
AD	Trade Capture Report Request	O
AE	Trade Capture Report	I

1.3.3 Supported and Unsupported Fields

This document only describes the FIX workflows and fields necessary to access FX Grid. All fields not included in this document are unsupported and ignored by FX Grid.

The Req'd column in the message tables indicates whether or not a field is required.

In some cases, a field is conditionally required or the FIX protocol requires fields that are not applicable to the FIX Client API and are therefore ignored by FX Grid. These fields are clearly indicated in their field comments.

1.3.4 String Length

For many fields of type `String`, the FIX protocol defines all possible valid values. Therefore, the maximum string length of these fields is also defined. The maximum string length never exceeds the longest valid value.

The FIX Client API does not impose a maximum length on undefined `String` fields, such as free-form text fields and ID fields.

1.3.5 Message Length

The FIX Client API does not impose a maximum length on FIX messages sent or received by FX Grid.

1.3.6 Encryption

For messages sent over the public Internet, the FIX Client API expects the message to be encrypted. If you use a secure network transport mechanism (VPN, Radianz, or SSL), the FIX messages may be unencrypted. You must work with your Integral relationship manager to establish the message encryption scheme.

1.3.7 Standard Header and Trailer

FIX engines set a message's header and trailer fields automatically according to the message type and the application context as defined by the application's configuration (see "Configuration" on page 54). For these reasons, the standard header and trailer as defined by the FIX protocol are not discussed in detail in this document beyond the expected values of the `MsgType` (#35) field and the

fields that identify message and business senders and targets, such as `SenderCompID` (#49), and `TargetCompID` (#56).

***Table 1-1** Standard Message Header Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
8	BeginString	Y	FIX.4.3	String	The identifier at the beginning of a new message that also holds the protocol version. Always set to "FIX.4.3". Always the first field in the message.
9	BodyLength	Y	—	Length	Indicates the message length in bytes. Always the second field in the message.
35	MsgType	Y	—	String	Defines the message type. Always the third field in the message. See "Supported Message Types" on page 50 for the complete list of supported message types.
34	MsgSeqNum	Y	—	SeqNum	This value is an integer message sequence number.
43	PossDupFlag	N	Y=Possible duplicate N=Original transmission	Boolean	Indicates possible retransmission of message with this sequence number.
49	SenderCompID	Y	—	String	The message sender's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	The legal entity of the message sender. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	The message target's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	See descr.	—	String	The legal entity of the message target. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	—	String	The ID of the message's business content originator. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. See "Users" on page 57 and "Summaries of ID Values" on page 58. When the stream LP trading workflow is used, the ID of the stream LP is sent in the Trade Capture Report in this field.
128	DeliverToCompID	See descr.	—	String	The ID of the message's business content target. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of the message's business content target. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
52	SendingTime	Y	—	UTCTimestamp	The time of message transmission (GMT) in the format *YYYYMMDD-hh:mm:ss.sss*.
122	OrigSendingTime	N	—	UTCTimestamp	Original time of message transmission when transmitting orders as the result of a resend request in the format *YYYYMMDD-hh:mm:ss.sss*.

***Table 1-2** Standard Message Trailer Field*

Tag	Field Name	Req'd	Value	FIX Format	Comments
10	Checksum	Y	—	String	A three byte, simple checksum that is always the last field on the message.

1.4 Configuration

The following sections described the details involved in establishing a connection via the FIX Client API.

1.4.1 Network Connectivity

The FIX Client API uses TCP/IP sockets. FIX message encryption is not supported. Clients connect to FX Grid using one of the following options to ensure network reliability and security:

- Radianz
- IPSec VPN
- Internet with SSL encryption using Stunnel

1.4.2 Organization and User Identification

The IDs used by the FIX Client API to identify message senders and targets must be received from Integral before you can trade with the FIX Client API through FX Grid.

For a quick summary of how the FIX Client API uses these IDs, see "Summaries of ID Values" on page 58.

The following sections describe these IDs and their values in detail:

Message Sender and Target

You must receive your organization ID and FX Grid server ID from Integral before you can connect to FX Grid using the FIX Client API. The FIX Client API uses these IDs to identify a message's sender and target in the `SenderCompID` (#49) and `TargetCompID` (#56) fields.

NOTE: FX Grid does not allow multiple FIX Client API connections using the same organization ID.

Your Organization Client ID

Your organization ID is expressed as a series of tokens separated by periods, much like an Internet domain name:

`sessionType.orgShortName`

`quote.yourBank4`

***Table 1-1** Organization ID Format*

Token	Example	Description
sessionType	**quote**.yourBank4	For a description of the following session types, see "Sessions" on page 11: ■ quote ■ order
orgShortName	quote.**yourBank4**	Your organization's short name ID

All messages that you send to FX Grid must include your ID in the SenderCompID (#49) field.

All messages that you receive from FX Grid include your ID in the TargetCompID (#56) field.

For a summary of how you should set the IDs of messages, see "Direct Users ID Summary Tables" on page 58 and "Facilitator Users ID Summary Tables" on page 61.

FX Grid Server ID

Like your organization ID, the FX Grid server ID looks like an Internet domain name:
environment.client.*otherData*
staging.client.fxgrid.integral.com

***Table 1-2** Organization ID Format*

Token	Example	Description
environment	**staging** .client.fxgrid.integral.com	The specific environment to which you are connected (for example, staging or production)
client	staging. **client**.fxgrid.integral.com	The FIX API (always client)
otherData	staging.client. **fxgrid.integral.com**	Additional data that identifies the server

All messages that you send to FX Grid must include the server ID in the TargetCompID (#56) field.

All messages that you receive from FX Grid include your ID in the SenderCompID (#49) field.

For a message-by-message summary of how you should set the IDs of messages, see "Direct Users ID Summary Tables" on page 58 and "Facilitator Users ID Summary Tables" on page 61.

Business Sender and Target

When you deal directly with liquidity providers, the message sender and target are the same as the business sender and target.

For example, if you send a New Order - Single message to initiate a trade, the SenderCompID (#49) field identifies you as both the message sender and the business sender (the organization initiating the trade). The message target is indicated by the value of the DeliverToCompID (#128) field, either the specific provider organization ID or blank if the order is intended to be matched with prices from all providers.

However, if you deal with liquidity providers on behalf of a customer, the business content of a message is sourced from or intended for your customer rather than you. The business sender and target is your customer. The FIX Client API uses the following fields to identify your customers as business senders and targets:

Table 1-3 *ID Fields for Facilitator Customers*

ID	Fields
Customer as Sender	■ Customer user/organization: `OnBehalfOfCompID` (#115) The user ID and organization is expressed in the form *user@custOrg*. If the field contains only the customer organization ID, the organization's default user is assumed. ■ Customer legal entity: `OnBehalfOfSubID` (#116) If the field is blank, the organization's default legal entity is assumed.
Customer as Target	■ Customer organization/user: `DeliverToCompID` (#128) The user ID and organization is expressed in the form *user@custOrg*. ■ Customer legal entity: `DeliverToSubID` (#129).

For example, if you send a New Order - Single message to initiate a trade on behalf of a customer, then you must set the `SenderCompID` (#49) field with your organization ID and then set the `OnBehalfOfCompID` (#115) field to the customer's ID, using either *user@custOrg* to specify the user and organization or just *custOrg* to assume the organization's default user.

You must receive these customer IDs from Integral before you can trade with the FIX Client API through FX Grid.

For a summary of how the FIX Client API uses these IDs, see "Direct Users ID Summary Tables" on page 58 and "Facilitator Users ID Summary Tables" on page 61.

See "Client Roles" on page 44 for a description of user roles.

Broadcast Messages

Whether you are a facilitator or a direct user, you can choose to send Market Data Request messages and New Order - Single messages to specific liquidity providers or to broadcast the messages to all subscribed providers.

To send a Market Data Request to a single provider or match a New Order - Single message with prices from a single provider, set the `DeliverToCompID` (#128) field with the provider organization's ID.

To match New Order - Single message with prices from a specific list of providers, set the `DeliverToCompID` (#128) field with a comma-separated list of provider organization IDs.

To broadcast a Market Data Request to all providers or match a New Order - Single message with prices from all subscribed providers, leave the `DeliverToCompID` (#128) field empty. Subscribed providers are providers who are currently streaming prices to the FX Grid and have a trading relationship with the message's business sender.

Users

For both direct customers and facilitators, your organization's trading user ID is sent in the `PartyID` (#448) field of all user-initiated messages. See "Client Roles" on page 44 if you are not sure whether you are a direct customer or a facilitator.

If you are a sales dealer or facilitator organization who trades with liquidity providers on behalf of a customer, your customer's user ID and organization is captured in the following fields and messages in the form *user@custOrg*:

***Table 1-4** Facilitator Customer User ID Fields and Messages*

Field	Messages
`OnBehalfOfCompID` (#115)	If the field contains only the customer organization ID, the organization's default user is assumed. ■ New Order – Single ■ Quote Request ■ Quote Cancel ■ Order Cancel Request ■ Order Cancel/Replace Request ■ Order Status Request ■ Order Mass Status Request
`DeliverToCompID` (#128)	Execution Report

You must receive these user IDs from Integral before you can trade with the FIX Client API through FX Grid.

For a summary of how the FIX Client API uses these IDs, see "Direct Users ID Summary Tables" on page 58 and "Facilitator Users ID Summary Tables" on page 61.

See "Client Roles" on page 44 for a description of user roles.

Legal Entities and Trading Parties

A *legal entity* is a sub-organization within your organization. Legal entities are normally regarded as having a unique legal existence. They produce balance sheets and report to central authorities.

A legal entity in an organization other than your own is referred to as a *trading party*.

Trading relationships are established between legal entities and trading parties.

The legal entity or trading party ID is contained in the following fields of several message types:

- `SenderSubID` (#50)
- `TargetSubID` (#57)
- `OnBehalfOfSubID` (#116): for facilitator users, this field may be left blank and the customer organization's default trading party is assumed for some messages. Additionally, when the stream LP trading workflow is used, the ID of the stream LP is sent in this field.
- `DeliverToSubID` (#129)

You must receive these IDs from Integral before you can trade with the FIX Client API through FX Grid.

For a complete summary of these rules, see "Direct Users ID Summary Tables" on page 58 and "Facilitator Users ID Summary Tables" on page 61.

Summaries of ID Values

The following tables summarize the ID values described in this section from the direct and facilitator perspectives. See "Client Roles" on page 44 for a description of direct and facilitator users.

- "Direct Users ID Summary Tables" on page 58
- "Facilitator Users ID Summary Tables" on page 61

Direct Users ID Summary Tables

The following tables summarizes the ID values in messages for direct users. If you are not certain that you are a direct customer, see "Client Roles" on page 44 for a detailed description of user roles and "Trading Workflows" on page 12 for more information about workflows.

***Table 1-5** Org and User IDs for Direct Users: Session Management*

FIX Field	Message Type				
	Logon	**Logout**	**Heartbeat**	**Test Request**	**Session-Level Reject**
`SenderCompID` (#49)	directional	directional	directional	directional	directional
`PartyID` (#448)	—	—	—	—	—
`TargetCompID` (#56)	directional	directional	directional	directional	directional
`MDEntryOriginator` (#282)	—	—	—	—	—
`DeliverToCompID` (#128)	—	—	—	—	—
`DeliverToSubID` (#129)	—	—	—	—	—
`OnBehalfOfCompID` (#115)	—	—	—	—	—
`OnBehalfOfSubID` (#116)	—	—	—	—	—
`SenderSubID` (#50)	—	—	—	—	—
`TargetSubID` (#57)	—	—	—	—	—

***Table 1-6** Org and User IDs for Direct Users: ESP Workflow*

FIX Field	Message Type				
	Market Data Request	**Market Data Snapshot/Full Refresh**	**Market Data Request Reject**	**New Order – Single**	**Execution Report**
`SenderCompID` (#49)	*clientID*	*serverID*	*serverID*	*clientID*	*serverID*
`PartyID` (#448)	—	—	—	*userID*	—
`TargetCompID` (#56)	*serverID*	*clientID*	*clientID*	*serverID*	*clientID*

Table 1-6 *Org and User IDs for Direct Users: ESP Workflow (continued)*

FIX Field	Message Type				
	Market Data Request	**Market Data Snapshot/Full Refresh**	**Market Data Request Reject**	**New Order – Single**	**Execution Report**
`MDEntryOriginator` (#282)	—	*providerOrg*	—	—	—
`DeliverToCompID` (#128)	*providerOrg*, ALL for aggregated market data from all providers, or blank for individual messages from all providers	—	—	*providerOrg*, list of providers, or blank for all providers	—
`DeliverToSubID` (#129)	—	—	—	—	—
`OnBehalfOfCompID` (#115)	—	—	*providerOrg* (if applicable)	—	*providerOrg*
`OnBehalfOfSubID` (#116)	—	—	—	—	—
`SenderSubID` (#50)	*legalEntity*	—	—	*legalEntity*	*providerLE*
`TargetSubID` (#57)	—	*legalEntity*	*legalEntity*	—	*legalEntity*

Table 1-7 *Org and User IDs for Direct Users: RFS Workflow*

FIX Field	Message Type				
	Quote Request	**Quote**	**Quote Request Reject**	**Quote Cancel**	**Business Message Reject**
`SenderCompID` (#49)	*clientID*	*serverID*	*serverID*	*clientID*	*serverID*
`PartyID` (#448)	—	—	—	—	—
`TargetCompID` (#56)	*serverID*	*clientID*	*clientID*	*serverID*	*clientID*
`MDEntryOriginator` (#282)	—	—	—	—	—
`DeliverToCompID` (#128)	*providerOrg*	—	—	*providerOrg*	—
`DeliverToSubID` (#129)	—	—	—	—	—
`OnBehalfOfCompID` (#115)	—	*providerOrg*	*providerOrg*	—	—
`OnBehalfOfSubID` (#116)	—	—	—	—	—
`SenderSubID` (#50)	*legalEntity*	—	—	*legalEntity*	—
`TargetSubID` (#57)	—	*legalEntity*	*legalEntity*	—	—

Table 1-8 *Org and User IDs for Direct Users: Order Workflow*

FIX Field	Message Type						
	Order Cancel Request	**Order Cancel/Replace Request**	**Order Cancel Reject**	**Order Mass Cancel Request**	**Order Mass Cancel Report**	**Order Status Request**	**Order Mass Status Request**
`SenderCompID` (#49)	*clientID*	*clientID*	*serverID*	*clientID*	*serverID*	*clientID*	*clientID*
`PartyID` (#448)	*userID*	*userID*	—	*userID*	—	*userID*	*userID*
`TargetCompID` (#56)	*serverID*	*serverID*	*clientID*	*serverID*	*clientID*	*serverID*	*serverID*

***Table 1-8** Org and User IDs for Direct Users: Order Workflow (continued)*

FIX Field	Message Type						
	Order Cancel Request	**Order Cancel/Replace Request**	**Order Cancel Reject**	**Order Mass Cancel Request**	**Order Mass Cancel Report**	**Order Status Request**	**Order Mass Status Request**
`MDEntryOriginator` (#282)	—	—	—	—	—	—	—
`DeliverToCompID` (#128)	*providerOrg*	*providerOrg*	—	*providerOrg*	—	*providerOrg*	*providerOrg*
`DeliverToSubID` (#129)	—	—	—	—	—	—	—
`OnBehalfOfCompID` (#115)	—	—	*providerOrg*	—	*providerOrg*	—	—
`OnBehalfOfSubID` (#116)	—	—	—	—	—	—	—
`SenderSubID` (#50)	*legalEntity*	*legalEntity*	—	*legalEntity*	—	*legalEntity*	*legalEntity*
`TargetSubID` (#57)	—	—	*legalEntity*	—	*legalEntity*	—	—

***Table 1-9** Org and User IDs for Direct Users: Trade Capture*

FIX Field	Message Type	
	Trade Capture Request	**Trade Capture Report**
`SenderCompID` (#49)	*clientID*	*serverID*
`PartyID` (#448)	—	—
`TargetCompID` (#56)	*serverID*	*clientID*
`MDEntryOriginator` (#282)	—	—
`DeliverToCompID` (#128)	—	*counterpartyA_Org*
`DeliverToSubID` (#129)	—	—
`OnBehalfOfCompID` (#115)	—	*counterpartyB_Org*
`OnBehalfOfSubID` (#116)	—	When the stream LP trading workflow is used, the ID of the stream LP is sent in this field.
`SenderSubID` (#50)	—	CptyA receives: *counterpartyA_LE* CptyB or other receives: *tradingPartyA*
`TargetSubID` (#57)	—	CptyB receives: *counterpartyB_LE* CptyA or other receives: *tradingPartyB*

Key:

- —: Do not include this field in the message.
- *clientID*: Your FIX client organization ID. See "Message Sender and Target" on page 54.
- *serverID*: The FX Grid server ID. See "Message Sender and Target" on page 54.
- directional: Either *clientID* or *serverID* depending on the message direction
- *legalEntity*: Your legal entity ID. See "Legal Entities and Trading Parties" on page 57.
- *providerOrg*: The liquidity provider organization's ID

- *providerLE*: The liquidity provider organization's legal entity or trading party ID. See "Legal Entities and Trading Parties" on page 57.
- *userID*: Your organization's trading user's ID
- *counterpartyA_Org*: Counterparty A organization ID (taker).
- *counterpartyB_Org*: Counterparty B organization ID (maker).
- *counterpartyA_LE*: Counterparty A legal entity ID (taker).
- *counterpartyB_LE*: Counterparty B legal entity ID (maker).
- *tradingPartyA*: Trading party ID or settlement code that represents Counterparty A's legal entity
- *tradingPartyB*: Trading party ID or settlement code that represents Counterparty B's legal entity

Facilitator Users ID Summary Tables

The following tables summarize the ID values in messages for facilitator users. If you are not certain that you are a facilitator, see "Client Roles" on page 44 for a detailed description of user roles.

***Table 1-10** Org and User IDs for Facilitator Users: Session Management*

FIX Field	Message Type				
	Logon	**Logout**	**Heartbeat**	**Test Request**	**Session-Level Reject**
`SenderCompID` (#49)	directional	directional	directional	directional	directional
`PartyID` (#448)	—	—	—	—	—
`TargetCompID` (#56)	directional	directional	directional	directional	directional
`MDEntryOriginator` (#282)	—	—	—	—	—
`DeliverToCompID` (#128)	—	—	—	—	—
`DeliverToSubID` (#129)	—	—	—	—	—
`OnBehalfOfCompID` (#115)	—	—	—	—	—
`OnBehalfOfSubID` (#116)	—	—	—	—	—
`SenderSubID` (#50)	—	—	—	—	—
`TargetSubID` (#57)	—	—	—	—	—

***Table 1-11** Organization and User IDs for Facilitator Users: ESP Workflow*

FIX Field	Message Type				
	Market Data Request	**Market Data Snapshot/Full Refresh**	**Market Data Request Reject**	**New Order – Single**	**Execution Report**
`SenderCompID` (#49)	*clientID*	*serverID*	*serverID*	*clientID*	*serverID*
`PartyID` (#448)	—	—	—	*userID*	—
`TargetCompID` (#56)	*serverID*	*clientID*	*clientID*	*serverID*	*clientID*
`MDEntryOriginator` (#282)	—	*providerOrg*	—	—	—

Table 1-11 *Organization and User IDs for Facilitator Users: ESP Workflow (continued)*

FIX Field	Message Type				
	Market Data Request	**Market Data Snapshot/Full Refresh**	**Market Data Request Reject**	**New Order – Single**	**Execution Report**
`DeliverToCompID` (#128)	*providerOrg*, ALL for aggregated market data from all providers, or blank for individual messages from all providers	—	—	*providerOrg*, list of providers, or blank for all providers	*user@custOrg*
`DeliverToSubID` (#129)	—	—	—	—	*customerTP*
`OnBehalfOfCompID` (#115)	—	—	*providerOrg* (if applicable)	*user@custOrg* or *custOrg* assuming default user	*providerOrg*
`OnBehalfOfSubID` (#116)	—	—	—	*customerLE* or blank to use default LE of *custOrg*	—
`SenderSubID` (#50)	*facilitatorLE*	—	—	—	*providerOrg*
`TargetSubID` (#57)	—	*facilitatorLE*	*facilitatorLE*	—	—

Table 1-12 *Organization and User IDs for Facilitator Users: RFS Workflow*

FIX Field	Message Type				
	Quote Request	**Quote**	**Quote Request Reject**	**Quote Cancel**	**Business Message Reject**
`SenderCompID` (#49)	*clientID*	*serverID*	*serverID*	*clientID*	*serverID*
`PartyID` (#448)	*userID*	—	—	*userID*	—
`TargetCompID` (#56)	*serverID*	*clientID*	*clientID*	*serverID*	*clientID*
`MDEntryOriginator` (#282)	—	—	—	—	—
`DeliverToCompID` (#128)	*providerOrg*	*user@custOrg*	*user@custOrg*	*providerOrg*	—
`DeliverToSubID` (#129)	—	*customerLE*	*customerLE*	—	—
`OnBehalfOfCompID` (#115)	*user@custOrg* or *custOrg* assuming default user	*providerOrg*	*providerOrg*	*user@custOrg* or *custOrg* assuming default user	—
`OnBehalfOfSubID` (#116)	*customerLE* or blank to use default LE of *custOrg*	—	—	*customerLE* or blank to use default LE of *custOrg*	—
`SenderSubID` (#50)	—	—	—	—	—
`TargetSubID` (#57)	—	—	—	—	—

Table 1-13 *Org and User IDs for Facilitator Users: Order Workflow*

FIX Field	Message Type						
	Order Cancel Request	**Order Cancel/Replace Request**	**Order Cancel Reject**	**Order Mass Cancel Request**	**Order Mass Cancel Report**	**Order Status Request**	**Order Mass Status Request**
`SenderCompID` (#49)	*clientID*	*clientID*	*serverID*	*clientID*	*serverID*	*clientID*	*clientID*
`PartyID` (#448)	*userID*	*userID*	—	*userID*	—	*userID*	*userID*

***Table 1-13** Org and User IDs for Facilitator Users: Order Workflow (continued)*

FIX Field	Message Type						
	Order Cancel Request	Order Cancel/Replace Request	Order Cancel Reject	Order Mass Cancel Request	Order Mass Cancel Report	Order Status Request	Order Mass Status Request
`TargetCompID` (#56)	*serverID*	*serverID*	*clientID*	*serverID*	*clientID*	*serverID*	*serverID*
`MDEntryOriginator` (#282)	—	—	—	—	—	—	—
`DeliverToCompID` (#128)	*providerOrg*	*providerOrg*	*user@custOrg*	*providerOrg*	*user@custOrg*	*providerOrg*	*providerOrg*
`DeliverToSubID` (#129)	—	—	*customerLE*	—	*customerLE*	—	—
`OnBehalfOfCompID` (#115)	*user@custOrg*	*user@custOrg*	*providerOrg*	—	*providerOrg*	*user@custOrg* or *custOrg* assuming default user	*user@custOrg*
`OnBehalfOfSubID` (#116)	*customerLE*	*customerLE*	—	*customerLE*	—	*customerLE* or blank to use default LE of *custOrg*	*customerTP*
`SenderSubID` (#50)	—	—	—	—	—	—	—
`TargetSubID` (#57)	—	—	—	—	—	—	—

***Table 1-14** Org and User IDs for Facilitator Users: Trade Capture*

FIX Field	Message Type	
	Trade Capture Request	Trade Capture Report
`SenderCompID` (#49)	*clientID*	*serverID*
`PartyID` (#448)	—	—
`TargetCompID` (#56)	*serverID*	*clientID*
`MDEntryOriginator` (#282)	—	—
`DeliverToCompID` (#128)	—	*counterpartyA_Org*
`DeliverToSubID` (#129)	—	—
`OnBehalfOfCompID` (#115)	—	*counterpartyB_Org*
`OnBehalfOfSubID` (#116)	—	When the stream LP trading workflow is used, the ID of the stream LP is sent in this field.
`SenderSubID` (#50)	—	CptyA receives: *counterpartyA_LE* CptyB or other receives: *tradingPartyA*
`TargetSubID` (#57)	—	CptyB receives: *counterpartyB_LE* CptyA or other receives: *tradingPartyB*

Key:

- —: Do not include the field in the message.
- *clientID*: Your FIX client organization ID. See "Message Sender and Target" on page 54.
- *serverID*: The FX Grid server ID. See "Message Sender and Target" on page 54.
- directional: Either *clientID* or *serverID* depending on the message direction.
- *user@custOrg*: Your customer's user and organization ID.

- *facilitatorLE*: Your legal entity ID. See "Legal Entities and Trading Parties" on page 57 and "Client Roles" on page 44.
- *customerTP*: Your customer's legal entity ID. See "Legal Entities and Trading Parties" on page 57 and "Client Roles" on page 44.
- *providerOrg*: The liquidity provider organization's ID.
- *providerLE*: The liquidity provider organization's legal entity or trading party ID. See "Legal Entities and Trading Parties" on page 57.
- *userID*: Your organization's trading user's ID
- *counterpartyA_Org*: Counterparty A organization ID (taker).
- *counterpartyB_Org*: Counterparty B organization ID (maker).
- *counterpartyA_LE*: Counterparty A legal entity ID (taker).
- *counterpartyB_LE*: Counterparty B legal entity ID (maker).
- *tradingPartyA*: Trading party ID or settlement code that represents Counterparty A's legal entity
- *tradingPartyB*: Trading party ID or settlement code that represents Counterparty B's legal entity

1.4.3 IP Address and Port

The provider initiates the connection to FX Grid, which specifies the IP address and port number. The provider's system sends a Logon message to the given IP and port. Communication starts with a Logon message and ends with a Logout message.

CHAPTER 2

Session Management

2.1 Session Management Messages

The following messages are used to control the FIX session and manage message conversations.

- "Logon" on page 65
- "Logout" on page 66
- "Heartbeat" on page 67
- "Test Request" on page 67
- "Session-Level Reject" on page 68

2.2 Logon

(Bidirectional)

The Logon message is sent by the client application to start a FIX session with FX Grid and sent by FX Grid in response.

If FX Grid receives a Logon message with invalid fields, it sends a Logout message in response. See "Logout" on page 66.

Table 2-1 *Logon Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Comments
35	MsgType	Y	A	String	A=Logon
49	SenderCompID	Y	—	String	ID of your organization sending the message. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	The message target. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

***Table 2-1** Logon Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Comments
98	EncryptMethod	Y	0	int	You must work with your Integral relationship manager to establish the message encryption scheme. For connectivity that provides inherent transport-level security (for example, Radianz or VPN): 0 (zero)=no encryption is used For messages delivered over the public Internet: 1 = PKCS (proprietary) 2 = DES (ECB mode) 3 = PKCS/DES (proprietary) 4 = PGP/DES (defunct) 5 = PGP/DES-MD5 (see note on FIX web site) 6 = PEM/DES-MD5
108	HeartBtInt	Y	—	int	Heartbeat interval in seconds. The heartbeat interval is driven by the FIX client. This value is set on the client-side `config.properties` file as `SERVER.POLLING.INTERVAL`. The value of this field should be 30 or greater. The default value is 30 seconds. If `HeartBtInt` is set to zero, then no heart beat message is required.
141	ResetSeqNumFlag	N	—	Boolean	Indicates that the both sides of the FIX session should reset sequence numbers. See "Sequence Number Reset" on page 43 for more information. Valid values: ■ Y=Yes, reset sequence numbers ■ N=No, continue with the current sequence number on the server until end-of-day.
553	Username	Y	—	String	The user's ID. This user is associated with the `SenderCompID` (#49) value, which is the user's organization.
554	Password	Y	—	String	The user's password.

2.3 Logout

(Bidirectional)

The Logout message is sent by the client application to end a session with FX Grid and sent by FX Grid in response.

If FX Grid receives a Logon message with invalid fields, it sends a Logout message in response with a description of the error in the `Text` (#58) field.

***Table 2-1** Logout Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Comments
35	MsgType	Y	5	String	5=Logout
49	SenderCompID	Y	—	String	The message sender. If outbound, the ID of your organization sending the message. If inbound, the FX Grid server ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	ID of the message target, either your organization ID or the FX Grid server. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

***Table 2-1** Logout Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Comments
58	Text	N	—	String	The reason for the Logon rejection. Only included for incoming Logout messages in response to an invalid Logon message. Not valid for outgoing Logout messages.

2.4 Heartbeat

(Bidirectional)

Both the client application and FX Grid send the Heartbeat message to indicate that the connection is active.

The provider's client 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.

***Table 2-1** Heartbeat Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Comments
35	MsgType	Y	0	String	0 (zero)=Heartbeat
49	SenderCompID	Y	—	String	The message sender. If outbound, the ID of your organization sending the message. If inbound, the FX Grid server ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	ID of the message target, either your organization ID or the FX Grid server. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
112	TestReqID	N	—	String	Required when the heartbeat is the result of a Test Request message. See "Test Request" on page 67.

2.5 Test Request

(Bidirectional)

Forces a heartbeat from the receiving system. The receiving system responds to a Test Request message with a Heartbeat message containing the `TestReqID`. See "Heartbeat" on page 67.

***Table 2-1** Test Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Comments
35	MsgType	Y	1	String	1=Test Request
49	SenderCompID	Y	—	Boolean	The message sender. If outbound, the ID of your organization sending the message. If inbound, the FX Grid server ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	ID of the message target, either your organization ID or the FX Grid server. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

Table 2-1 *Test Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Comments
112	TestReqID	Y	—	String	The resulting Heartbeat message contains this ID. The `TestReqID` should increment.

2.6 Session-Level Reject

(Bidirectional)

Sent when a message is received but cannot be properly processed due to a session-level rule violation. See the "SessionRejectReason" field for a list of rejection reasons.

Rejected messages should be logged and the incoming sequence number incremented.

Table 2-1 *Session-Level Reject Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Comments
35	MsgType	Y	3	String	3=Reject
45	RefSeqNum	Y	—	SeqNum	The referenced message's sequence number.
58	Text	N	—	String	Free format text string
354	EncodedTextLen	N	—	Length	Byte length of encoded (non-ASCII characters) EncodedText field.
355	EncodedText	N	—	data	Encoded (non-ASCII characters) representation of the `Text` (#58) field in the encoded format specified via the `MessageEncoding` (#347) field in the standard header. If used, the ASCII (English) representation should also be specified in the `Text` (#58) field.
371	RefTagID	N	—	int	The tag number of the FIX field that caused the message to be rejected.
372	RefMsgType	N	—	String	The `MsgType` (#35) of the FIX message being referenced.

Table 2-1 *Session-Level Reject Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Comments
373	SessionRejectReason	N	—	int	The coded reason for the rejection. Valid values: ■ 0=Invalid tag number ■ 1=Required tag missing ■ 2=Tag not defined for this message type ■ 3=Undefined Tag ■ 4=Tag specified without a value ■ 5=Value is incorrect (out of range) for this tag ■ 6=Incorrect data format for value ■ 7=Decryption problem ■ 8=Signature problem ■ 9=CompID problem ■ 10=SendingTime accuracy problem ■ 11=Invalid MsgType ■ 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 NumInGroup count for repeating group ■ 17=Non "data" value includes field delimiter (SOH character)

CHAPTER 3

Market Data and Quote Workflow

3.1 Supported Actions

The FIX Client API offers the following actions to FIX clients:

- Market data request for the ESP workflow
- Quote request for the RFS workflow

3.2 Supported Message Types

FX Grid order workflow supports the following FIX messages types:

Table 3-1 *Supported Message Types*

Message Type—MsgType (#35) Field Value	Message Name	Workflow	Inbound to Client (I), Outbound from Client (O), or Bidirectional (B)
V	Market Data Request	ESP	O
W	Market Data Snapshot/Full Refresh	ESP	I
Y	Market Data Request Reject	ESP	I
R	Quote Request	RFS	O
AG	Quote Request Reject	RFS	I
S	Quote	RFS	I
Z	Quote Cancel	RFS	I

3.3 Market Data Messages

The messages in this section are used to access the ESP workflow of FX Grid (see "Executable Streaming Prices (ESP) Workflow" on page 12).

3.3.1 Market Data Request

(ESP Workflow Only: Client to FX Grid)

The client must submit a Market Data Request message for each currency pair from each liquidity provider to initiate trading in the ESP workflow (see "Executable Streaming Prices (ESP) Workflow" on page 12).

- "Request Types" on page 71
- "Message Fields" on page 72

Request Types

You can request prices:

- From specific providers or all of your providers
- As full refresh (snapshot)
- As single prices or aggregated
- As full book, best price, or price tiers

The following table describes the typical combinations of price request attributes:

Table 3-1 *Typical Price Requests*

LPs	Subscription Type	Market Depth	Aggregation	Price Message	Market Data Request Field Values
All LPs	Full/Snapshot	Full book	Full book	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=0 (full book) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=F
All LPs	Full/Snapshot	Best price	Full book	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=1 (Top of Book) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=F
All LPs	Full/Snapshot	Price tiers	Full book	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=N>1 (Best N price tiers of data) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=F

Table 3-1 *Typical Price Requests (continued)*

LPs	Subscription Type	Market Depth	Aggregation	Price Message	Market Data Request Field Values
All LPs	Full/Snapshot	Full book	VWAP	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=0 (Full Book) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=V
All LPs	Full/Snapshot	Best price	VWAP	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=1 (Top of Book) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=V
All LPs	Full/Snapshot	Price tiers	VWAP	Market Data Snapshot/Full Refresh	■ `DeliverToCompID` (#128)=ALL ■ `MDUpdateType` (#265)=0 (full refresh) ■ `MarketDepth` (#264)=N>1 (Best N price tiers of data) ■ `AggregatedBook` (#266)=Y ■ `AggregationType` (#7548)=V

Message Fields

The Market Data Request message consists of the following fields:

Table 3-2 *Market Data Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	V	String	V=Market Data Request
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	Your legal entity's ID if you are trading directly with a liquidity provider. If you are trading on behalf of a customer, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

***Table 3-2** Market Data Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
128	DeliverToCompID	Y	■ "ALL" for one message for all providers combined ■ Provider ID for specific provider ■ Comma-separated list for one message from select providers combined ■ Empty for one message for each provider separately	String	The liquidity provider organization's ID if the message is intended for a specific provider. To receive prices from all providers in a single combined message, set this field to "ALL". You can then specify whether you want the prices merged or not with the `AggregatedBook` (#266) field. To receive prices only from select providers in a single combined message, set this field to a comma-separated list of provider IDs. You can then specify whether you want the prices merged or not with the `AggregatedBook` (#266) field. If this field is left empty, then you subscribe to all providers and receive a separate market data message for each provider. Instead of this approach, Integral recommends that you establish separate FIX sessions for each liquidity provider. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
262	MDReqID	Y	—	String	The client-assigned unique ID for the market data request. This is stored in the external request ID field. FX Grid rejects all requests with duplicate IDs. The value of this field must not contain the ampersand character "@".
263	SubscriptionRequestType	Y	1=Snapshot + Updates (subscribe to stream, default) 2=Disable previous Snapshot + Update Request (Unsubscribe)	char	This field indicates to the receiving party what type of response is expected. A subscribe request asks for updates as the status changes. Unsubscribe cancels any future update messages from the organization.
264	MarketDepth	Y	■ 0=Full book ■ 1=Top of book ■ *N* > 1=Best *N* price tiers of data	int	Depth of market for book snapshot.
265	MDUpdateType	See descr.	0=Full Refresh	int	This field is required when the `SubscriptionRequestType` (#263) field is set to 1 (Snapshot + Updates). This field specifies the type of Market Data update.

Table 3-2 *Market Data Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
266	AggregatedBook	N	N=Multiple entries per side per price allowed Y=One book entry per side per price	Boolean	Specifies whether or not book entries should be merged. For an aggregated full book, this field value must be "Y" and `DeliverToCompID` (#128) must be set to "ALL". If `AggregationType` (#7548)=V (VWAP), then prices are sent aggregated in a single market data message as a multi-tier quote. See "Quote Types" on page 45 for information about multi-price and multi-tier quotes. If this field is set to "Y" and `AggregationType` (#7548) is not specified, then full book aggregation is done. When this field value is set to "Y", then `MDEntryOriginator` (#282) on the merged prices is determined by how the system is configured, how the price tiers are defined, and provider priority. See "Provider Names with Merged Prices and Provider Priority" on page 45. When this field value is set to "N", then `MDEntryOriginator` (#282) on each price that you receive is the provider organization's ID.
7546	RequestedSize	N	—	Qty	Specifies the size of the tier. If multiple tiers are required, use commas to separate tier values. The following size shortcuts are supported: K, k, M, m, B, b.
7548	AggregationType	N	F=Full Book V=VWAP	MultipleValueString	The type of aggregation to be done. If this tag is not specified and `AggregatedBook` (#266)=Y, Full Book aggregation is done. `DeliverToCompID` (#128) must be set to "ALL" to receive an aggregated full book. If `AggregationType` (#7548)=V (VWAP) then prices are sent aggregated in a single market data message as a multi-tier quote. See "Quote Types" on page 45 for information about multi-price and multi-tier quotes.
The following shaded rows are a repeating group of fields that represent one currency pair. The required fields (Symbol (#55) and Product (#460)) must be included as a group for each currency pair or your request will be rejected. The value of the NoRelatedSym (#146) field indicates the number of groups and thus the number of currency pairs.					
146	NoRelatedSym	Y	1 (one)	NumInGroup	The number of repeating symbols specified. This indicates the number of currency pairs that the market data request message is associated with. Because the client must submit a Market Data Request Message for each currency pair, there is a one-to-one relationship between the quote request message and currency pair. Thus, the value of this field is always `1` (one).
55	Symbol	Y	—	String	The symbol for the base and term currencies of the currency pair in the following format: *baseCCY/termCCY* (for example, "EUR/USD")

***Table 3-2** Market Data Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
15	Currency	N	—	Currency	The dealt currency of the request. This may be the base or term currency of a currency pair. If the term currency: ■ Only prices from providers who support term currency trades are considered for aggregation. ■ The requested amount is converted to the base currency to determine appropriate tier from the multi-tier rate. The mid point of the top-of-book price is used to perform the conversion. ■ The `MinQty` (#110) amount is converted to the base currency before it is used to test the requested order size. ■ If the Market Data Request specifies multiple tiers in the `RequestedSize` (#7546) field, the tier sizes are interpreted in term currency. The tier sizes are then converted to the base currency and used for aggregation.
460	Product	Y	4	int	The asset class. The value is always 4=CURRENCY.
167	SecurityType	N	FOR	String	The security type. The value is always "FOR"=Foreign Exchange Contract
110	MinQty	N	—	Qty	When specified, the value of this field is enforced for aggregation. A quote with a size smaller than this value is not included in an aggregated price.
End of repeating group: NoRelatedSym (#146)					
The following shaded rows are repeating fields required by the FIX specification. The value of the NoMDEntryTypes (#267) field is always 2. Two instances of the MDEntryType (#269) are always required and always have the value 0 and 1 respectively.					
267	NoMDEntryTypes	Y	2	NumInGroup	Number of `MDEntryType` (#269) fields requested. This number is always set to "2" (bid/offer).
269	MDEntryType	Y	(see descr.) ■ 0 (zero) ■ 1 (one)	char	The FIX Client API supports only two-way market data. This field is required by the FIX specification but is ignored by the FIX Client API. There must be two instances of the `MDEntryType` (#269) field. One instance is set to "0" (zero) and one instance is set to "1" (one).
End of repeating group: NoMDEntryTypes (#267)					

3.3.2 Market Data Snapshot/Full Refresh

(ESP Workflow Only: FX Grid to Client)

FX Grid sends a Market Data Snapshot/Full Refresh message to the client in response to a successful Market Data Request message for each currency pair from each liquidity provider. The Market Data Snapshot/Full Refresh message may contain multiple rates. The `NoMDEntries` (#268) field indicates the number of rates in the message. This message is only used in the ESP workflow (see "Executable Streaming Prices (ESP) Workflow" on page 12).

Each rate consists of a repeating group of fields, as indicated by shading in the table below.

Table 3-1 *Market Data Snapshot/Full Refresh Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	W	String	W=Market Data Snapshot/Full Refresh
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	Y	—	String	Your legal entity's ID. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	N	—	String	Not included in the message. The ID of the originating provider is captured in the `MDEntryOriginator` (#282) field of each quote in the message. See "MDEntryOriginator" on page 77.
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
262	MDReqID	Y	—	String	Unique identifier from the originating Market Data Request
460	Product	Y	4	int	The asset class. The value is always 4=CURRENCY.
541	MaturityDate	Y	—	LocalMktDate	Date of maturity in *YYYYMMDD* format
The following shaded rows are a repeating group of fields that represent one rate. The required fields must be included as a group for each rate. The value of the NoMDEntries (#268) field indicates the number of groups and thus the number of rates.					
268	NoMDEntries	Y	—	NumInGroup	Number of entries in market data message. Each bid and offer represents one market data entry. If three bid and two offer dealing prices are included, the value of the `NoMDEntries` (#268) field is `5`.
269	MDEntryType	Y	—	char	The side of the rate: 0=Bid 1=Offer/Ask
270	MDEntryPx	Y	—	Price	The price. For example, if `MDEntryType` (#269) field of a repeating group is `0` (bid), this field holds the bid price. If the value of this field is 0 (zero), your client should ignore the quote.
15	Currency	Y	—	Currency	The value of this field represents the denomination of the quantity fields (for example, `JPY` represents a quantity of JPY). This may be the base or term currency of a currency pair. If the term currency: ■ Only prices from providers who support term currency trades are considered for aggregation. ■ The requested amount is converted to the base currency to determine appropriate tier from the multi-tier rate. The mid point of the top-of-book price is used to perform the conversion. ■ The `MinQty` (#110) amount is converted to the base currency before it is used to test the requested order size. ■ If the Market Data Request specifies multiple tiers in the `RequestedSize` (#7546) field, the tier sizes are interpreted in term currency. The tier sizes are then converted to the base currency and used for aggregation.

Table 3-1 *Market Data Snapshot/Full Refresh Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
271	MDEntrySize	Y	—	Qty	The quantity (in the case of multiple tiers, the limit). If the value of this field is 0 (zero), your client should ignore the quote.
276	QuoteCondition	Y	—	MultipleValueString	Indicates whether a stream's prices are tradable or not: A=Open/Active (stream is active and last quote is tradable) B=Closed/Inactive (stream is inactive and the last quote is not tradable)
282	MDEntryOriginator	Y	—	String	The provider organization ID. The provider associated with the bid or offer quote. If `AggregatedBook` (#266)=Y on the originating market data request, the value of this field on the merged prices is determined by how the system is configured, how the price tiers are defined, and provider priority. See "Provider Names with Merged Prices and Provider Priority" on page 45. If `AggregatedBook` (#266)=N on the originating market data request, the value of this field is the provider's ID.
110	MinQty	N	—	Qty	When specified in the subscription request, the value of this field is enforced for aggregation. A quote with a size smaller than the value of this field is not included in an aggregated price.
299	QuoteEntryID	Y	—	String	Uniquely identifies each rate as part of a quote set. The reference ID for the dealing price.
290	MDEntryPositionNo	Y	—	int	The integer value indicates the tier of the price. If a price is part of a multi-price quote and is not part of a multi-tier quote, this field has a value of 0 (zero). See "Quote Types" on page 45 for information about business rules regarding multi-price and multi-tier quotes.
End of repeating group: NoMDEntries (#268)					

3.3.3 Market Data Request Reject

(ESP Workflow Only: FX Grid to Client)

FX Grid sends a Market Data Request Reject message to the client in response to an unsuccessful Market Data Request message for each currency pair. This message is only used in the ESP workflow (see "Executable Streaming Prices (ESP) Workflow" on page 12).

Table 3-1 *Market Data Request Reject Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	Y	String	Y=Market Data Request Reject
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	Y	—	String	Your legal entity's ID. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.

Table 3-1 *Market Data Request Reject Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
115	OnBehalfOfCompID		—	String	■ Not included if the originating request was intended for all liquidity providers ■ The liquidity provider organization's ID if the originating request was intended for a specific provider. See the "DeliverToCompID" field under "Market Data Request" on page 71.
58	Text	N	—	String	A description of the rejection
262	MDReqID	Y	—	String	The unique identifier from the originating Market Data Request. Since the request is stored with the external request ID, the external request ID is stored with the quote (dealing prices) message.
281	MDReqRejReason	N	—	char	The reason for the reject. If the client attempts to subscribe to a currency pair that they are not permissioned for, the value of the `MDReqRejReason` (#281) field is `3`. 0=Unknown symbol 1=Duplicate `MDReqID` 3=Insufficient Permissions

3.4 Quote Messages

The messages in this section are used to access the RFS workflow of FX Grid (see "Request for Stream (RFS) Workflow" on page 17).

3.4.1 Quote Request

(RFS Workflow Online: Client to FX Grid)

The client sends a Quote Request message to FX Grid to initiate the RFS workflow (See "Request for Stream (RFS) Workflow" on page 17).

The "Batch" column in the table indicates fields that are required on Quote Request messages for batch trades:

- Y: Required for batch trades
- N: Not applicable. Ignored if included on a Quote Request for batch trades.
- See descr.: Conditionally required for batch trades. The field description includes the conditional information.
- —: Optional

***Table 3-1** Quote Request Fields*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
35	MsgType	Y	Y	R	String	R=Quote Request
49	SenderCompID	Y	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	—	String	Your legal entity's ID if you are trading directly with a liquidity provider. If you are trading on behalf of a customer, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	Y	—	String	The ID of the liquidity provider organization to whom the request is sent. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
448	PartyID	See descr.	—	—	String	The ID of your organization's user who submitted the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Users" on page 57 and "Summaries of ID Values" on page 58.
1	Account	N	See descr.	—	String	For batch trades, this field is required if the batch is allocated to a single account.
131	QuoteReqID	Y	Y	—	String	Client-assigned unique ID for the quote request. For batch trades, this ID is included in all responses from FX Grid.

***Table 3-1** Quote Request Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
146	NoRelatedSym	Y	Y	1	NumInGroup	Number of related instruments in the request. This value is always "1" (one).
55	Symbol	Y	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
167	SecurityType	Y	Y	FOR	String	FOR=Foreign Exchange Contract
54	Side	See descr.	See descr.	1=Buy 2=Sell Space=2-Way 7=Undisclosed	Char	The order side is from customer's (FIX client) perspective. For FX swap, it is the side of the far leg. Undisclosed batch (two-way price): Required with value "7".
38	OrderQty	Y	See descr.	—	Qty	Requested amount specified in the dealt currency `Currency` (#15). For FX swap, this is the near leg amount. Dislosed batch (one-way price): This field is not applicable and must not be included on the request for dislcosed (one-way price) batch quote. Undisclosed batch (two-way price): Required.
64	FutSettDate	N	N	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	■ FX spot: Spot date ■ FX outright: Value date ■ FX swap (spot-fwd and fwd-fwd): Near leg value date The field contains either a standard tenor symbol or broken date in *YYYYMMDD* format for outright and swap. If the field is empty, it is considered a spot stream request.
40	OrdType	Y	N	—	Char	D=Previously quoted (for FX spot RFS, FX outright RFS, and NDF) G=FX swap
193	FutSettDate2	N	N	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either standard tenor symbol or a broken date in *YYYYMMDD* format.
192	OrderQty2	N	N	—	Qty	Requested amount specified in the dealt currency `Currency` (#15). For FX swap, this is the far leg amount.
126	ExpireTime	Y	—	—	UTCTimestamp	The relative expiration time of the quote request in the format *YYYYMMDD-hh:mm:ss.sss*. The date portion of the value is ignored. The time portion specifies the amount of time before the quote request expires. For example, for a request that expires in ten seconds, the value would be 00000000-00:00:10.000.
60	TransactTime	Y	—	—	UTCTimestamp	The time the Quote Request is sent by the client. This field is automatically stamped by the FIX engine in the format *YYYYMMDD-hh:mm:ss.sss*.
15	Currency	Y	N	—	Currency	Dealt currency

Table 3-1 *Quote Request Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
541	MaturityDate	See descr.	N	—	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64). NDF swaps: The near-leg fixing date
8541	FarMaturityDate	See descr.	N	—	LocalMktDate	For NDF swaps only: The far-leg fixing date
9374	USINamespace	N	—	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9375	USIFarNamespace	N	—	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9376	USI	N	—	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	—	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9378	UTIPrefix	N	N	—	String	UTI Namespace for single-leg trades (outrights) and near leg of swaps
9379	UTIFarPrefix	N	N	—	String	UTI Namespace for far leg of swaps. Not applicable to single-leg trades.
9380	UTI	N	N	—	String	UTI value for single-leg trades (outrights) and near leg of swaps
9381	UTIFar	N	N	—	String	UTI value for far leg of swaps. Not applicable to single-leg trades.
The following shaded rows are a repeating group of fields that represent a trade leg in the batch of trades. The value of the NoLegs (#555) field indicates the number of groups and thus the number of legs.						
555	NoLegs	Y	Y	—	NumInGroup	For batch trades: Repeating group for multi-leg trade details
600	LegSymbol	Y	Y	—	String	The leg's currency pair
624	LegSide	Y	Y	—	char	The side of the leg: ■ 1=Buy ■ 2=Sell ■ 7=Undisclosed (for undisclosed batch trades)
556	LegCurrency	N	Y	—	Currency	The dealt currency of the leg
687	LegQty	Y	Y	—	Qty	The dealt amount of the leg. For undisclosed batch requests, this value is the net amount for given value date. For example, if two legs for 1W value date with amount 1M buy and 2M sell, then the value of this field for 1W will be 1M. The side of the leg is only revealed after price acceptance.
588	LegSettDate	Y	Y	—	LocalMktDate	Value date of the leg

***Table 3-1** Quote Request Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
The following shaded rows are a repeating group of fields that represent a sub account for allocations per leg. The value of the NoNestedPartyIDs (#539) field indicates the number of groups and thus the number of sub accounts.						
539	NoNestedPartyIDs	Y	Y	—	NumInGroup	The number of sub accounts for pre-trade allocation of the leg
524	NestedPartyID		See descr.	—	String	The organization ID of the sub account for pre-trade allocations only. Batch trades: Not required for undisclosed batch. Required for disclosed batch.
End of repeating group: NoNestedPartyIDs (#539)						
654	LegRefID		Y	—	String	The unique ID of the trade represented by the leg in the batch. For multi-leg trades, such as swap trades, this ID should be the same for all legs of the trade. This element must be the last element in each `NoLegs` (#555) group.
9378	UTIPrefix	N	N	—	String	UTI Namespace
9380	UTI	N	N	—	String	UTI value
End of repeating group: NoLegs (#555)						

3.4.2 Quote Request Reject

(RFS Workflow Online: FX Grid to Client)

FX Grid sends a Quote Request Reject message to the client to reject a previously sent Quote Request. This message is only used in the RFS workflow (see "Request for Stream (RFS) Workflow" on page 17).

***Table 3-1** Quote Request Reject Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	AG	String	AG=Quote Request Reject
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	See descr.	—	String	■ The originating legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer
115	OnBehalfOfCompID	Y	—	String	The liquidity provider organization's ID. See "Summaries of ID Values" on page 58.
128	DeliverToCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.

***Table 3-1** Quote Request Reject Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
58	Text	N	—	String	Free format text explaining the reason for rejection. If the `Text` (#58) field includes the text "INTERNAL_SERVER_ERROR", this indicates a serious error. Stop trading immediately and contact Business Support to resolve the issue.
131	QuoteReqID	Y	—	String	The `OrdReqID` (#131) of the Quote Request message that has been rejected
658	QuoteRequestRejectReason	Y	—	String	The reason for the rejection: ■ 1=Unknown symbol ■ 2=Exchange or security closed ■ 3=Quote request exceeds limit ■ 4=Not authorized to request quote ■ 99=Other
The following shaded rows are a repeating group of fields that represent instruments.					
146	NoRelatedSym	Y	—	NumInGroup	Number of related instruments in the request. For batch trades, the value of this field can be greater than one.
60	TransactTime	Y	—	UTCTimestamp	The time the Quote Request reject message is generated in the format *YYYYMMDD-hh:mm:ss.sss*.
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
End of repeating group: NoRelatedSym (#146)					

3.4.3 Quote

(RFS Workflow Online: FX Grid to Client)

FX Grid sends a Quote message in response to a successful Quote Request. This message is only used in the RFS workflow (see "Request for Stream (RFS) Workflow" on page 17).

The "Batch" column in the table indicates if the field is sent with batch Quotes:

- Y: Included for batch trades
- N: Not included on the message for batch trades
- —: Optional

***Table 3-1** Quote Fields*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
35	MsgType	Y	Y	S	String	S=Quote
49	SenderCompID	Y	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

Table 3-1 *Quote Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
57	TargetSubID	See descr.	—	—	String	■ The originating legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer
128	DeliverToCompID	See descr.	—	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	Y	—	—	String	The liquidity provider organization's ID. See "Summaries of ID Values" on page 58.
15	Currency	N	N	—	Currency	Dealt currency
40	OrdType	Y	—	—	Char	D=Previously quoted (for FX spot and FX outright RFS. G=FX swap
54	Side	N	N	1=Buy 2=Sell Empty=2-Way	Char	The order side is from customer's (FIX client) perspective. For FX swap, it is the side of the far leg. Not included in batch quotes. Each trade in the batch specifies its side.
55	Symbol	Y	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
60	TransactTime	Y	—	—	UTCTimestamp	The time the message is generated. This field is automatically stamped by the FIX engine in the format *YYYYMMDD-hh:mm:ss.sss*.
62	ValidUntilTime	N	—	—	UTCTimestamp	Expiry time of the quote in the format *YYYYMMDD-hh:mm:ss.sss*. Only the time portion is used as a time duration. For example, 00:01:30.000 means the quote is good for 1 minute and 30 seconds.
64	FutSettDate	N	N	—	LocalMktDate	The field is specified in the format *YYYYMMDD*. ■ FX spot: Spot date ■ FX outright: Outright value date ■ FX spot-forward swap: Spot date ■ FX forward-forward swap: Near value date
117	QuoteID	Y	Y	—	String	Unique quote ID assigned by FX Grid
131	QuoteReqID	Y	Y	—	String	Client-assigned unique ID for the quote request, taken from the `QuoteReqID` (#131) on the originating Quote Request message. For batch trades, this ID is included in all responses from FX Grid.

Table 3-1 *Quote Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
132	BidPx	N	N	—	Price	All-in bid price. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way. ■ FX spot: Spot price ■ FX outright : Outright price ■ FX spot-forward swap: Spot price ■ FX forward-forward swap: Near leg all-in price If the value of this field is 0 (zero), your client should ignore the quote.
133	OfferPx	N	N	—	Price	All-in offer price. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way. If the value of this field is 0 (zero), your client should ignore the quote.
134	BidSize	N	N	—	Qty	Bid size of the near leg quote. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way. If the value of this field is 0 (zero), your client should ignore the quote.
135	OfferSize	N	N	—	Qty	Offer size of the near leg quote. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way. If the value of this field is 0 (zero), your client should ignore the quote.
167	SecurityType	Y	—	FOR	String	FOR=Foreign Exchange Contract
188	BidSpotRate	N	N	—	Price	Bid spot rate. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way. If the value of this field is 0 (zero), your client should ignore the quote.
189	BidForwardPoints	N	N	—	PriceOffset	Bid forward points. The value is used for outrights and fwd/fwd swaps. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way.
190	OfferSpotRate	N	N	—	Price	Offer spot rate. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way. If the value of this field is 0 (zero), your client should ignore the quote.
191	OfferForwardPoints	N	N	—	PriceOffset	Offer forward points. The value is used for outrights and fwd/fwd swaps. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way.
193	FutSettDate2	N	N	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either standard tenor symbol or a broken date in *YYYYMMDD* format.
537	QuoteType	See descr.	Y	0 (zero) =Indicative (the quote is not tradable) 1=Tradable	int	Whether or not the quote is tradable. Quotes for batch trades are always tradable (1).
541	MaturityDate	See descr.	N	—	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64). NDF swaps: The near-leg fixing date

Table 3-1 *Quote Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
8541	FarMaturityDate	See descr.	N	—	LocalMktDate	For NDF swaps only: The far-leg fixing date
631	MidPx	N	N	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	N	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
642	BidForwardPoints2	N	N	—	PriceOffset	Bid Forward Points for the far leg. The value is used for Swap and Fwd/Fwd Swap only. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way.
643	OfferForwardPoints2	N	N	—	PriceOffset	Offer Forward Points for the far leg. The value is used for Swap and Fwd/Fwd Swap only. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way.
7551	BidSize2	N	N	—	Qty	Bid size of the far leg quote. Included only if `Side` (#54) in the originating Quote Request is sell or 2-way.
7552	OfferSize2	N	N	—	Qty	Offer size of the far leg quote. Included only if `Side` (#54) in the originating Quote Request is buy or 2-way.
The following shaded rows are a repeating group of fields that represent a trade leg in the batch of trades. The value of the NoLegs (#555) field indicates the number of groups and thus the number of legs.						
555	NoLegs	Y	Y	—	NumInGroup	Repeating group for batch quotes
600	LegSymbol	Y	Y	—	String	The leg's currency pair
624	LegSide	Y	Y	—	char	The side of the leg: ■ 1=Buy ■ 2=Sell ■ 7=Undisclosed (for undisclosed batch trades)
556	LegCurrency	N	Y	—	Currency	The dealt currency of the leg
588	LegSettDate	N	Y	—	LocalMktDate	Value date of the leg
681	LegBidPx	N	Y	—	Price	Bid spot rate.
684	LegOfferPx	N	Y	—	Price	Offer spot rate.
654	LegRefID		Y	—	String	The unique ID of the trade represented by the leg in the batch. For multi-leg trades, such as swap trades, this ID should be the same for all legs of the trade.
1067	LegBidFwdPoints	N	Y	—	PriceOffset	Bid forward points. For example, if the forward points for EUR/USD using market conventions, for any given tenor, is 8 then we expect to receive 0.0008 instead of 8.
1068	LegOfferFwdPoints	N	Y	—	PriceOffset	Offer forward points. For example, if the forward points for EUR/USD using market conventions, for any given tenor, is 8 then we expect to receive 0.0008 instead of 8.
7632	MidPxLeg	N	N	—	Price	For SEF trades: The mid rate

Table 3-1 *Quote Fields (continued)*

Tag	Field Name	Req'd	Batch	Value	FIX Format	Description
End of repeating group: NoLegs (#555)						

3.4.4 Quote Cancel

(RFS Workflow Online: Bidirectional)

FX Grid sends a Quote Cancel message to the client to stop an RFS quote stream and cancel all quotes.

The client sends a Quote Cancel message to FX Grid as part of the RFS workflow to cancel a Quote Request message and stop an RFS stream. See "Request for Stream (RFS) Workflow" on page 17.

Table 3-1 *Quote Cancel Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	Z	String	Z=Quote Cancel
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	Your legal entity's ID if you are trading directly with a liquidity provider. If you are trading on behalf of a customer, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
448	PartyID	See descr.	—	String	The ID of your organization's user who submitted the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Users" on page 57 and "Summaries of ID Values" on page 58.
60	TransactTime	Y	—	UTCTimestamp	The time the message was generated. This field is automatically stamped by the FIX engine in the format *YYYYMMDD-hh:mm:ss.sss*.
117	QuoteID	See descr.	* (asterisk) for batch trades	String	This field is required when the message is sent by the FX Grid. This field is not required when the message is sent by the customer to cancel a quote request. For batch trades, the value is * (asterisk) when the message is sent by the FX Grid to cancel all published quotes.
131	QuoteReqID	Y	—	String	The `QuoteReqID` (#131) of the originating quote request.

Table 3-1 *Quote Cancel Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
298	QuoteCancelType	Y	—	int	1=Cancel for Symbols, used only for batch trades 4=Quote Withdraw, used when the quote stream is canceled by the provider 5=Quote/Request Expired, used when the request or quote expiry is reached

CHAPTER 4

Trading Workflow

4.1 Supported Actions

The FIX Client API offers the following actions to FIX clients:

- Order submission
- Order cancel/replace
- Order status query
- Order execution reports for fills, order submission failure, and queries
- Position management
- Post-trade STP download

4.2 Supported Message Types

FX Grid order workflow supports the following FIX messages types:

Table 4-1 *Supported Message Types*

Message Type—MsgType (#35) Field Value	Message Name	Inbound to Client (I), Outbound from Client (O), or Bidirectional (B)
D	New Order – Single	O
AB	New Order – Multileg	O
F	Order Cancel Request	O
G	Order Cancel/Replace Request	O
9	Order Cancel Reject	I
q	Order Mass Cancel Request	O
r	Order Mass Cancel Report	I
H	Order Status Request	O
AF	Order Mass Status Request	O
8	Execution Report	I
j	Business Message Reject	B

***Table 4-1** Supported Message Types (continued)*

Message Type—MsgType (#35) Field Value	Message Name	Inbound to Client (I), Outbound from Client (O), or Bidirectional (B)
AN	Request for Positions	O
AO	Request for Positions Ack	I
AP	Positions Report	I
AD	Trade Capture Report Request	O
AE	Trade Capture Report	I

4.3 Trading Messages

The messages in this section are used to access the trading workflow of FX Grid for all trading workflows (See "Trading Workflows" on page 12).

4.3.1 New Order – Single

(Client to FX Grid)

The client sends a New Order – Single message to submit an order in FX Grid.

The order can be an execution request on a streaming price (`OrdType` (#40)=D) or a limit order (`OrdType` (#40)=2) that is crossed with quotes on the server and broadcast to other market participants. See "Trading Workflows" on page 12 for more information about trading workflows.

For RiskNet directed orders, the New Order – Single message from the client must contain the following fields and values:

- `REXParameters` (#7700)=TIV~<Integer value in seconds> (for example, TIV~10)
- `ExecInst` (#18)=VN B (Venue and OK to cross when `REXParameters` (#7700)=TIV expires)
- `DeliverToCompID` (#128)=Trading venue ID. If `REXParameters` (#7700) is specified and this field is not, a validation error is returned.

***Table 4-1** New Order – Single Message Fields*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
35	MsgType	Y	—	D	String	D=Submit one new order
43	PossDupFlag	N	—	Y=Possible duplicate N=Original transmission	Boolean	Must not be set. See "Duplicate Order IDs" on page 34.
122	OrigSendingTime	Y (if #43 is set)	—	See descr.	UTCTimestamp	Original time of message transmission in the format *YYYYMMDD-hh:mm:ss.sss*. Required if the order message is a resend (`PossDupFlag` (#43) field is set to "Y").

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
49	SenderCompID	Y	—	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	___	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider.or for RiskNet directed orders. See "Legal Entities and Trading Parties" on page 57 and "Message Sender and Target" on page 54 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.
56	TargetCompID	Y	___	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	___	—	String	Your customer organization's ID if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	___	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
128	DeliverToCompID	See descr.	___	—	String	The liquidity provider organization's ID or a comma-separated list of provider IDs to match the order against. This field is required only for previously quoted orders (`OrdType` (#40) =D). For limit, market, and stop orders, this field is optional. If this field is included, then only the providers specified are considered for matching with the order. If this field is empty or not included, FX Grid attempts to match the order against prices from all subscribed providers. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58. Required for RiskNet: The trading venue name.
11	ClOrdID	Y	___	—	String	A session-scoped unique identifier assigned by the client. If the client sends multiple New Order – Single messages for the same order, the same `ClOrdID` (#11) must be sent with each attempt. See "Duplicate Order IDs" on page 34.
15	Currency	Y	___	—	Currency	The dealt currency. This may be the base or term currency of a currency pair.
18	ExecInst	Y	___	B=OK to cross (execute at market) B not specified=No cross P=Best price P not specified=Price at depth 9=Bid trigger 0=Offer trigger M=Mid trigger R=At rate ST=Algo W=VWAP VN=Venue	MultipleValueString	How your order is executed. As applicable to the order type, you can specify more than one value by separating each value with a space. For details about how this field applies to the various order types, see "Order Execution" on page 30. Required for RiskNet: "VN" and "B" both to indicate that the order is directed to a specific venue (as specified by `DeliverToCompID` (#128)=Trading venue ID) and that the order can executed at market if the order is not completely filled within the time specified by `REXParameters` (#7700)='TIV~<number of seconds>.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
21	HandlInst	See descr.	___	1	char	1=Automated execution order, private, no manual intervention. Required by the FIX protocol but not applicable to the FIX Client API. 3=Manual order, best execution. Used for OMS integration. See "Integration with an Order Management System (OMS)" on page 42.
38	OrderQty	Y	___	—	Qty	FX spot, outright: The amount of the dealt currency (as specified by the `Currency` (#15) field) to be either bought or sold (as determined by the `Side` (#54) field). FX swap: The near leg currency amount of the dealt currency (as specified by the `Currency` (#15) field) to be either bought or sold (as determined by the `Side` (#54) field).
40	OrdType	Y	___	■ 1=Market or Market Range ■ 2=Limit (orders) ■ 3=Stop ■ 4=Stop Limit ■ D=Previously quoted (ESP, RFS, or NDF)	char	Other order types are not currently supported. See "Supported Order Types" on page 27. If the order type is D, then the `DeliverToCompID` (#128) field must have a value. If the order type is 1 or 3, then the value of the `PegOffsetValue` (#211) field determines whether the order is a market order or a market range order. See "PegOffsetValue" on page 97.
44	Price	See descr.	___	—	Price	This field is not required for market orders (OrdType (#40) =1). FX spot: The execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. FX swap: The near-leg execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. Stop limit orders: The limit price

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
54	Side	Y	___	1=Buy (Bid) 2=Sell (Offer)	char	The side of the order from the customer's perspective. For FX swap orders, this is the side of the far leg. If you are dealing directly with the liquidity provider, the customer is defined by the `SenderCompID` (#49) and `SenderSubID` (#50) fields. If you are dealing with the liquidity provider as a facilitator on behalf of the customer, the customer is defined by the `OnBehalfOfCompID` (#115) and `OnBehalfOfSubID` (#116) fields.
55	Symbol	Y	___	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
59	TimeInForce	Y	___	■ 0 (zero)=Day ■ 1=GTC ■ 3=IOC ■ 4=FOK ■ 6=GTT	char	See "Order Expiry" on page 31 for a description of order expiry types.
60	TransactTime	Y	___	—	UTCTimestamp	Time this order request was initiated/released by the trader, trading system, or intermediary in the format *YYYYMMDD-hh:mm:ss.sss*.
64	FutSettDate	N	RFS	—	LocalMktDate	Specific date of trade settlement (settlement date) in the format *YYYYMMDD*.
99	StopPx	N	RFS	—	Price	The stop price that triggers a stop or stop limit order. You can use the `ExecInst` (#18) field to specify whether the stop triggers when the market price equals or is less/greater than the stop price, or only when less/greater than the stop price. See "ExecInst" on page 92 and "Order Execution" on page 30 for details.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
110	MinQty	Y	RFS	■ 0 (zero) ■ Less than `OrderQty` (#38) ■ Equal to `OrderQty` (#38)	Qty	This field is ignored if `TimeInForce` (#59) is 4 (FOK). Specifies how the order can be filled: ■ Partial fill: If the value is 0 (zero), the order amount can be filled multiple times with any size until the entire amount is filled. ■ Partial fill with market minimum: If the value is less than the value of the `OrderQty` (#38) field, the first fill must be equal to or greater than the value of this field. Subsequent fills can be of any size. If the residual unfilled order amount is less than the value of this field, then the order is considered fully filled and done. The residual amount is implicitly canceled by the server. ■ No partial fill: If the value is equal to the value of the `OrderQty` (#38) field, the order amount must be filled in its entirety with exactly one fill. The value of this field also can determine whether or not the order is a displayed order. If an order's minimum fill (`MinQty` (#110)) is greater than its show amount (`MaxShow` (#210)), then the order is hidden and not displayed to other market participants. The order is only broadcast after the minimum fill amount is achieved. Conversely, if `MinQty` (#110) is less than `MaxShow` (#210), the displayed order is always broadcast. For this behavior to apply, the order in question must allow partial fills (`MinQty` (#110) is less than `OrderQty` (#38)) and the order must not have an FOK time in force (`TimeInForce` (#59) is not 4 (FOK)).
117	QuoteID	Y	RFS	—	String	The reference ID of the bid or offer dealing price. This is conditionally required, as the `OrdType` (#40) field is `D` (previously quoted).

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
126	ExpireTime	N	—	—	UTCTimestamp	The relative expiration time of the order in the format *YYYYMMDD-hh:mm:ss.sss*. Required when the `TimeInForce` (#59) value is 6 (GTT). The date portion of the value is ignored. The time portion specifies the amount of time before the order expires. For example, for an order that expires in ten seconds, the value could be 00000000-00:00:10.000.
167	SecurityType	Y	—	FOR	String	FOR=Foreign Exchange Contract
168	EffectiveTime	N	—	—	UTCTimestamp	The order submission time in GMT in the format *YYYYMMDD-hh:mm:ss* or *YYYYMMDD-hh:mm:ss.sss*. For algo orders, the value of this field indicates the absolute time in GMT at which the algo should start execution. If this field is not specified and `ExecEffPeriod` (#7564) is not specified, the algo starts execution immediately. If this field is specified and the time value is in the past, the algo starts execution immediately.
192	OrderQty2	N	RFS	—	Qty	FX spot, outright: Not applicable. FX swap: The far leg currency amount of the dealt currency (as specified by the `Currency` (#15) field) to be either bought or sold (as determined by the `Side` (#54) field).
193	FutSettDate2	N	RFS	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either standard tenor symbol or a broken date in *YYYYMMDD* format.
210	MaxShow	N	—	■ 0 (zero): hidden ■ Equal to `OrderQty` (#38): displayed ■ Less than `OrderQty` (#38): iceberg	Qty	See "Order Visibility" on page 33 for a description of order visibility. This field must either be included in the message with a value or left out of the message completely. If the field is included with no value, the message will be rejected.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
211	PegOffsetValue	See descr.	—	—	float	This field is required for: ■ Market range orders to specify the allowable slippage. The value is expressed in pips. For example, the value 1.0 equals 1 pip, which EUR/USD is 0.0001. ■ Stop orders if the resulting order when triggered is a market range order. This field is left empty or not included for market orders.
460	Product	Y	—	4	int	The asset class. The value is always 4=CURRENCY.
7549	OrderNotes	N	—	—	String	Notes on the order entered by the user who submitted the order. This field is only included on messages that are sent by the user who submitted the order or sent to the user's organization. Maximum 256 characters. Angle brackets "<" and ">" are not allowed in the value of this field.
541	MaturityDate	See descr.	RFS	N	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64). NDF swaps: The near-leg fixing date
8541	FarMaturityDate	See descr.	RFS	N	LocalMktDate	For NDF swaps only: The far-leg fixing date
583	ClOrdLinkID	N	—	—	String	The IDs of the orders to which this order is linked. This field is only populated in a New Order – Single message when submitting a component order of an OCO or OUO order chain. The value should be the `ClOrdID` (#11) values of the other component orders. See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
631	MidPx	N	SEF	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	SEF	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
640	Price2	N	RFS	—	Price	FX spot, outright: Not applicable. FX swap: The far-leg execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. The precision of this value must be agreed upon by both the sending and receiving parties.
1385	ContingencyType	N	—	See descr.	int	Defines the type of contingency. 1=One Cancels the Other (OCO) 3=One Updates the Other (OUO) - Absolute Quantity Reduction 4=One Updates the Other (OUO) - Proportional Quantity Reduction See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.
9360	IsSEFTrade	N	SEF	—	String	Y= The trade is a SEF Trade. N = All non-SEF NDF trades. Not applicable for other trade types. Ignored for outright/swap/spot trades.
9361	ExecutionVenue	N	SEF	—	String	For SEF trades LEI of the Integral SEF. For Non-SEF trades LEI of Integral.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
9362	USPerson	N	SEF	■ Y ■ N	String	Is the counterparty a "U.S. person" as defined by CTFC guidelines? Applicable to SEF trades only.
9363	SwapDataRepository	N	SEF	—	String	LEI of the SDR
9364	ClearingVenue	N	SEF	—	String	If this trade is a cleared SEF trade, LEI of clearing house
9365	TakerLEI	N	SEF	—	String	LEI of the taker counterparty
9366	MakerLEI	N	SEF	—	String	Default LEI of the maker counterparty. If the trade is booked with different LEI, please send new maker LEI in execution report. The alternate maker LEI should be pre-configured on Integral's system.
9367	IntentToClear	N	SEF	■ Y ■ N	String	Y= Intent to clear. N=Permitted SEF. Applicable for SEF trades only
9368	EndUserException	N	SEF	—	String	If the counterparty has invoked clearing exemption. Applicable to SEF trades only
9369	IsBlockTrade	N	SEF	■ Y ■ N	String	Is the trade a block trade? Applicable to SEF trades only.
9370	ReportingParty	N	SEF	■ Y ■ N	String	Y=the liquidity provider (maker) the reporting party. If the provider is a U.S. person as defined by CTFC guildelines (`USPerson` (#9363)=Y), reporting party is mandatory. N='The provider is not the reporting party.
9374	USINamespace	N	SEF	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9375	USIFarNamespace	N	SEF	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
9376	USI	N	SEF	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	SEF	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9378	UTIPrefix	N	SEF	—	String	UTI Namespace for single-leg trades (outrights) and near leg of swaps
9379	UTIFarPrefix	N	SEF	—	String	UTI Namespace for far leg of swaps. Not applicable to single-leg trades.
9380	UTI	N	SEF	—	String	UTI value for single-leg trades (outrights) and near leg of swaps
9381	UTIFar	N	SEF	—	String	UTI value for far leg of swaps. Not applicable to single-leg trades.
9382	UPI	N	SEF	—	String	UPI value
7556	ExecEndTime	See descr.	Algo	—	UTCTimestamp	Absolute time in GMT at which the algo should stop executing. If this tag is not specified and `ExecEndPeriod` (#7565) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.
The following shaded rows are key/value pairs specified in the AlgoParameters (#7560) field that represent an algo order's parameters. Each parameter key/value pair is delimited by a space. The tilde character (~) separates the parameter key from the parameter value. For example, the value "SI~01:00:00 SMI~00:30:00 SIR~Y AE~1" indicates "Slice interval = 1 hour, Minimum slice interval = 30 minutes, Slice interval randomized between 30 minutes (SMI) and 1 hour (SI), At expiration fill the remainder at market".						
7560	AlgoParameters	See descr.	Algo	—	MultipleValueString	The algo parameters represented as key/value pairs with the tilde character (~) separating the parameter key from the parameter value and each key/value pair delimited by a space. This field is required when `ExecInst` (#18) =ST.
AE	ActnOrdrExpr	N	Algo	1=Fill @ Market 2=Cancel Order	char	Action to be taken at the expiration of the order. If this tag is not specified, the order is cancelled at expiration.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
POT	PegOffsetType	N	Algo	1=Primary 2=Market 3=Mid 4=Transition	char	Indicates that the algo is a pegged order and specifies the peg offset type: ■ Primary: Bid if buying, offer if selling ■ Market: Bid if selling, offer if buying ■ Mid: Midpoint between the bid and the offer ■ Transition: First pegged to primary, then over the offset increment interval (`PII`, `PegOffsetIncrIntrvl`), moves to the market peg in increments by the offset increment (`POI`, `PegOffsetIncr`).
PO	PegOffset	N	Algo	—	Price	The amount in pips that is added to the price of the peg. Can be positive or negative.
POI	PegOffsetIncr	N	Algo	—	Price	The fixed amount in pips used to adjust the price offset for transition pegged orders
PII	PegOffsetIncrIntrvl	N	Algo	—	int	The interval at which the offset increment is applied for transition pegged orders. Specified in milliseconds.
PIR	PegOffsetIncrRndmzr	N	Algo	■ Y ■ N	Boolean	Whether or not the offset increment is randomized for transition pegged orders. When the offset increment is randomized, the increment value is a random value between zero and the offset increment (`POI`, `PegOffsetIncr`).
PT	PegTime	N	Algo	—	float	Peg time in seconds. For example, ten and a half seconds is represented as "10.5".
SFOK	SlcFillOrKill	N	Algo	Y=FOK N=IOC (default)	Boolean	Determines how slices are executed. Y=Slices must be filled completely by a single fill from a provider or cancelled. This parameter is used only if the slice size (`SS`, `SlcSize`) is fixed. N=Partial fills allowed.
SI	SlcIntrvl	Y	Algo	—	UTCTimeOnly	Relative time between two slices specified in the format *hh:mm:ss* [*.sss*]. This field is required when `ExecInst` (#18)=ST.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
SIR	SlcIntrvlRndmzr	N	Algo	■ Y ■ N	Boolean	Randomization enabled (Y) /disabled (N). If enabled, the order management system randomly selects a slice interval between `SlicIntrvl` (`SI`) and `SlcMinIntrvl` (`SMI`).
SMI	SlcMinIntrvl	Y	Algo	—	UTCTimeOnly	Minimum slice interval specified in the format *hh:mm:ss*[*.sss*]. Applicable if the `SlicIntrvlRndmzr` (`SIR`) parameter is true. This field is required when `ExecInst` (#18) =ST.
SRS	SlcRegularSize	N	Algo	—	Qty	If this parameter is specified, the slice size is rounded to a multiple of this value. If not specified, the behavior defaults to the system configuration. Contact your Integral Solutions Manager to configure the default slice size.
SS	SlcSize	N	Algo	—	Qty	Size of each slice. If the slice size is not specified, the size is chosen based on the `TOBPercent` (`TOBP`) parameter.
SSR	SlcSizeRndmzr	N	Algo	■ Y ■ N	Boolean	Randomization enabled (Y) /disabled (N) flag. If enabled, the order management system randomizes the slice size between `SlcRegularSize` (`SRS`) and `SlcSize` (`SS`) or, if `TOBPercent` (`TOBP`) is specified, between the calculated top-of-book amount and `SlcSize` (`SS`).
TOBP	TOBPercent	N	Algo	—	float	Determines how the slice size varies, calculated as a percentage of the top-of-book (`TOB`) size. For example, a value of 0.5 indicates 50% of TOB size. A value of 1.2 indicates 120% of TOB size. If `SlcSizeRndmzr` (`SSR`) is true, then the slice size varies between `SlcRegularSize` (`SRS`) and the size calculated with the value of this parameter. If this parameter is not specified, the slice size is fixed.
End of field parameters: AlgoParameters (#7560)						
7561	AlgoName	N	Algo	—	String	This field is used with algo orders (`ExecInst` (#18)=ST). A free-format string provided to record the algo employed by the order. The value of this field is not validated and has no effect on algo functionality.

Table 4-1 *New Order – Single Message Fields (continued)*

Tag	Field Name	Req'd	Workflow	Value	FIX Format	Description
7564	ExecEffPeriod		Algo	—	UTCTimeOnly	Relative time at which the algo should start execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `EffectiveTime` (#168) is not specified, the algo starts executing immediately.
7565	ExecEndPeriod		Algo		UTCTimeOnly	Relative time at which the algo should stop execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `ExecEndTime` (#7556) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.
7700	REXParameters	See descr.	—	TIV~<integer value in seconds>	String	Required for RiskNet directed orders: TIV (Time in Venue), the time in the trading venue specified in seconds. Should be less than `TimeInForce` (#59).
The following shaded rows are a repeating group of fields that represent a user. The required fields must be included as a group for each user. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of users.						
453	NoPartyIDs	N	—	1	NumInGroup	The number of groups of `PartyID` (#448), `PartyIDSource` (#447) and `PartyRole` (#452) fields that represent the end-user ID who submitted the order.
448	PartyID	N	—	—	String	The ID of the user who submitted the order. See "Users" on page 57 and "Summaries of ID Values" on page 58.
447	PartyIDSource	See descr.	—	D=Proprietary/Custom code	char	Identifies class or source of the `PartyID` (#448) value. Required if `PartyID` is specified.
452	PartyRole	N	—	3=Client ID	int	Identifies the type or role of the `PartyID` (#448) specified.
End of repeating group: NoPartyIDs (#453)						

4.3.2 New Order – Multileg

(Batch Workflow Only: Client to FX Grid)

The client sends a New Order – Multileg message to submit a order in FX Grid in the batch trading worflow (see "Batch Trade Workflow" on page 18).

Table 4-1 *New Order – Multileg Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	AB	String	AB=New Order – Multileg

Table 4-1 *New Order – Multileg Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
11	ClOrdID	Y	—	String	A session-scoped unique identifier assigned by the client. The value of this field must be the same as the `QuoteReqID` (#131) on the originating Quote Request and Quote messages. If the client sends multiple New Order – Multileg messages for the same order, the same `ClOrdID` (#11) must be sent with each attempt.
40	OrdType	Y	D=Previously quoted	char	The order type
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
60	TransactTime	Y	—		The time the message is sent by the client. This field is automatically stamped by the FIX engine in the format *YYYYMMDD-hh:mm:ss.sss*
75	TradeDate	Y	—	LocalMktDate	The trade date in the format *YYYYMMDD*
131	QuoteReqID	N	—	String	Client-assigned unique ID for the quote request
460	Product	Y	4	int	The asset class. The value is always 4=CURRENCY.
The following shaded rows are a repeating group of fields that represent a trade leg in the batch of trades. The value of the NoLegs (#555) field indicates the number of groups and thus the number of legs. The order of fields in the repeating group is different from the same repeating group in the Quote Request message.					
555	NoLegs	Y	—	NumInGroup	Repeating group for multi-leg trade details. The order of fields in the repeating group is different from the same repeating group in the Quote Request message.
600	LegSymbol	Y	—	String	The leg's currency pair
624	LegSide	Y	—	char	The side of the leg: Empty=2-way 1=Buy 2=Sell
556	LegCurrency	Y	—	Currency	The dealt currency of the leg
The following shaded rows are a repeating group of fields that represent a sub account for pre-trade allocations. The value of the NoNestedPartyIDs (#539) field indicates the number of groups and thus the number of sub accounts.					
539	NoNestedPartyIDs	N	—	NumInGroup	The number of repeating groups of allocations for the leg, if any
524	NestedPartyID		—	String	The organization ID of the sub account for pre-trade allocations only
End of repeating group: NoNestedPartyIDs (#539)					
654	LegRefID	Y	—	String	The unique ID of the trade represented by the leg in the batch. For multi-leg trades, such as swap trades, this ID should be the same for all legs of the trade. This element must be the last element in each NoLegs (#555) group.
588	LegSettDate	Y	—	LocalMktDate	Value date of the leg
9378	UTIPrefix	N	—	String	UTI Namespace
9380	UTI	N	—	String	UTI value
End of repeating group: NoLegs (#555)					

4.3.3 Order Cancel Request

(Client to FX Grid)

The client sends this message to FX Grid to request that a specific order be canceled.

Table 4-1 *Order Cancel Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	F	String	F=Order Cancel Request
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider. See "Message Sender and Target" on page 54 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
11	ClOrdID	Y	—	String	A unique ID for this cancel request assigned by the FIX client
41	OrigClOrdID	Y	—	String	The `ClOrdID` (#11) value of the order to be canceled
54	Side	Y	1=Buy (Bid) 2=Sell (Offer)	char	The side of the order from the FIX client's perspective
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
60	TransactTime	Y	—	UTCTimestamp	Time this cancel request was initiated/released by the FIX client in the format *YYYYMMDD-hh:mm:ss.sss*.
460	Product	Y	4	int	The asset class. The value is always 4=CURRENCY.
The following shaded rows are a repeating group of fields that represent a user. The required fields must be included as a group for each user. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of users.					

***Table 4-1** Order Cancel Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
453	NoPartyIDs	N	1	NumInGroup	The number of groups of `PartyID` (#448), `PartyIDSource` (#447) and `PartyRole` (#452) fields that represent the end-user ID who submitted the cancel request
448	PartyID	N	—	String	The ID of the user who submitted the cancel request. See "Users" on page 57 and "Summaries of ID Values" on page 58.
447	PartyIDSource	See descr.	D=Proprietary/Custom code	char	Identifies class or source of the `PartyID` (#448) value. Required if `PartyID` is specified.
452	PartyRole	N	3=Client ID	int	Identifies the type or role of the `PartyID` (#448) specified.
End of repeating group: NoPartyIDs (#453)					

4.3.4 Order Cancel/Replace Request

(Client to FX Grid)

The client sends this message to FX Grid to request that a specific order be canceled and then replaced with the order contained in the cancel/replace message.

> **NOTE:** Depending on how your system is configured and on the state of the order, the original order may be cancelled or amended. See "Order Cancel/Replace and Amendment" on page 28 for more information.

Groups of associated, repeating fields are indicated by shading in the table below.

***Table 4-1** Order Cancel/Replace Request Message Fields*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
35	MsgType	Y	—	G	String	G=Order Cancel/Replace Request
43	PossDupFlag	N	—	Y=Possible duplicate N=Original transmission	Boolean	Must not be set. See "Duplicate Order IDs" on page 34.
122	OrigSendingTime	Y (if #43 is set)	—	See descr.	UTCTimestamp	Original time of message transmission in the format *YYYYMMDD-hh:mm:ss.sss*. Required if the order message is a resend (`PossDupFlag` (#43) field is set to "Y").
49	SenderCompID	Y	—	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider. See "Message Sender and Target" on page 54 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
56	TargetCompID	Y	—	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	—	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58. Required for RiskNet: The trading venue name.
11	ClOrdID	Y	—		String	The order ID assigned by the FIX client for the replacement order
15	Currency	Y	—	—	Currency	The value of the Currency field represents the denomination of the quantity fields (for example, `JPY` represents a quantity of JPY). This may be the base or term currency of a currency pair.
18	ExecInst	N	—	B=OK to cross B not specified=No cross P=Best price P not specified=Price at depth 9=Bid trigger 0=Offer trigger M=Mid trigger R=At rate ST=Algo W=VWAP VN=Venue	MultipleValueString	How your order is executed. As applicable to the order type, you can specify more than one value by separating each value with a space. For details about how this field applies to the various order types, see "Order Execution" on page 30. If `ExecInst` (#18) is omitted from the Order Cancel/Replace Request, the value of `ExecInst` (#18) on the replacement order is set to "null", superseding any prior definition on the original order. If you want to persist the `ExecInst` (#18) settings of the original order, you must submit the original settings with the Order Cancel/Replace Request. Required for RiskNet: "VN B" both to indicate that the order is directed to a venue and that the order is matched if it is not filled before the time in the venue expires `REXParameters` (#7700)='TIV.
38	OrderQty	See descr.	—	—	Qty	The amount of the base currency to be either bought or sold (as determined by `Side` (#54) field). Required for all workflows except RiskNet.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
40	OrdType	Y	—	■ 1=Market or Market Range ■ 2=Limit (orders) ■ 3=Stop ■ 4=Stop Limit ■ D=Previously quoted (ESP, RFS, or NDF)	char	Other order types are not currently supported. See "Supported Order Types" on page 27. If the order type is 1, then the value of the `PegOffsetValue` (#211) field determines whether the order is a market order or a market range order. See "PegOffsetValue" on page 97.
44	Price	See descr.	—	—	Price	This field is not required for market orders (OrdType (#40)=1). FX spot: The execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. FX swap: The near-leg execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. Stop limit orders: The limit price
41	OrigClOrdID	Y	—		String	The `ClOrdID` (#11) value of the order to be canceled and replaced
54	Side	Y	—	1=Buy (Bid) 2=Sell (Offer)	char	The side of the order from the FIX client's perspective
55	Symbol	Y	—	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
59	TimeInForce	Y	—	■ 0 (zero)=Day ■ 1=GTC ■ 3=IOC ■ 4=FOK ■ 6=GTT	char	See "Order Expiry" on page 31 for a description of order expiry types.
58	Text	N	—		String	Free format text string
60	TransactTime	Y	—	—	UTCTimestamp	Time this order request was initiated/released by the trader, trading system, or intermediary in the format *YYYYMMDD-hh:mm:ss.sss*.
64	FutSettDate	N	—	—	LocalMktDate	Specific date of trade settlement (settlement date) in the format *YYYYMMDD*.
99	StopPx	N	—	—	Price	The stop price that triggers a stop or stop limit order. You can use the `ExecInst` (#18) field to specify whether the stop triggers when the market price equals or is less/greater than the stop price, or only when less/greater than the stop price. See "ExecInst" on page 107 and "Order Execution" on page 30 for details.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
110	MinQty	Y	—	■ 0 (zero) ■ Less than `OrderQty` (#38) ■ Equal to `OrderQty` (#38)	Qty	This field is ignored if `TimeInForce` (#59) is 4 (FOK). Specifies how the order can be filled: ■ Partial fill: If the value is 0 (zero), the order amount can be filled multiple times with any size until the entire amount is filled. ■ Partial fill with market minimum: If the value is less than the value of the `OrderQty` (#38) field, the first fill must be equal to or greater than the value of this field. Subsequent fills can be of any size. If the residual unfilled order amount is less than the value of this field, then the order is considered fully filled and done. The residual amount is implicitly canceled by the server. ■ No partial fill: If the value is equal to the value of the `OrderQty` (#38) field, the order amount must be filled in its entirety with exactly one fill. The value of this field also can determine whether or not the order is a displayed order. If an order's minimum fill (`MinQty` (#110)) is greater than its show amount (`MaxShow` (#210)), then the order is hidden and not displayed to other market participants. The order is only broadcast after the minimum fill amount is achieved. Conversely, if `MinQty` (#110) is less than `MaxShow` (#210), the displayed order is always broadcast. For this behavior to apply, the order in question must allow partial fills (`MinQty` (#110) is less than `OrderQty` (#38)) and the order must not have an FOK time in force (`TimeInForce` (#59) is not 4 (FOK)).
117	QuoteID	Y	—	—	String	The reference ID of the bid or offer dealing price. This is conditionally required, as the `OrdType` (#40) field is `D` (previously quoted).
126	ExpireTime	N	—	—	UTCTimestamp	The relative expiration time of the new order in the format *YYYYMMDD-hh:mm:ss.sss*. Required when the `TimeInForce` (#59) value is 6 (GTD). The date portion of the value is ignored. The time portion specifies the amount of time before the order expires. For example, for an order that expires in ten seconds, the value could be 00000000-00:00:10.000.
167	SecurityType	Y	—	FOR	String	FOR=Foreign Exchange Contract

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
168	EffectiveTime	N	—	—	UTCTimestamp	The order submission time in GMT in the format *YYYYMMDD-hh:mm:ss* or *YYYYMMDD-hh:mm:ss.sss*. For algo orders, the value of this field indicates, he absolute time in GMT at which the algo should start execution. If this field is not specified and `ExecEffPeriod` (#7564) is not specified, the algo starts execution immediately. If this field is specified and the time value is in the past, the algo starts execution immediately.
192	OrderQty2	N	—	—	Qty	FX spot, outright: Not applicable. FX swap: The far leg currency amount of the dealt currency (as specified by the `Currency` (#15) field) to be either bought or sold (as determined by the `Side` (#54) field).
193	FutSettDate2	N	—	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either standard tenor symbol or a broken date in *YYYYMMDD* format.
210	MaxShow	N	—	■ Empty ■ Equal to `OrderQty` (#38) ■ Less than `OrderQty` (#38)	Qty	The amount of the order visible to other market participants based on the value of the field: ■ Hidden: If the value is 0 (zero), empty, or not included, the order is a hidden order that is not visible to other customers. ■ Display: If the value is equal to the value of the `OrderQty` (#38) field, the order is a display order that is completely visible by other customers. The full order amount is disclosed. ■ Iceberg: If the value of is less than the value of the `OrderQty` (#38) field, the order is an iceberg order that is visible to other customers. Only a fraction of the actual order amount is displayed (the value of this field).
211	PegOffsetValue	See descr.	—	—	float	This field is required for: ■ Market range orders to specify the allowable slippage. The value is expressed in pips. For example, the value 1.0 equals 1 pip, which EUR/USD is 0.0001. ■ Stop orders if the resulting order when triggered is a market range order. This field is left empty or not included for market orders.
460	Product	Y	—	4	int	The asset class. The value is always 4=CURRENCY.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
7549	OrderNotes	N	—	—	String	Notes on the order entered by the user who submitted the order. This field is only included on messages that are sent by the user who submitted the order or sent to the user's organization. Maximum 256 characters. Angle brackets "<" and ">" are not allowed in the value of this field.
541	MaturityDate	See descr.	—	N	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64). NDF swaps: The near-leg fixing date
8541	FarMaturityDate	See descr.	—	N	LocalMktDate	For NDF swaps only: The far-leg fixing date
583	ClOrdLinkID	N	—	—	String	The IDs of the orders to which this order is linked. This field is only populated in a New Order – Single message when submitting a component order of an OCO or OUO order chain. The value should be the `ClOrdID` (#11) values of the other component orders. See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.
631	MidPx	N	—	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	—	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
640	Price2	N	—	—	Price	FX spot: Not applicable. FX swap: The far-leg execution price or limit price. The precision of this float value must be agreed upon by both the sending and the receiving parties. The precision of this value must be agreed upon by both the sending and receiving parties.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
1385	ContingencyType	N	—	See descr.	int	Defines the type of contingency. 1=One Cancels the Other (OCO) 3=One Updates the Other (OUO) - Absolute Quantity Reduction 4=One Updates the Other (OUO) - Proportional Quantity Reduction See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.
9360	IsSEFTrade	N	—	—	String	Y - if the trade is a SEF Trade. N - for all Non SEF NDF trades. Not applicable for other trade types – Outright / swap / Spot to be ignored)
9361	ExecutionVenue	N	—	—	String	For SEF trades LEI of the Integral SEF. For Non-SEF trades LEI of Integral.
9362	USPerson	N	—	—	String	Y/N - Is the counterparty a US Person
9363	SwapDataRepository	N	—	—	String	LEI of the SDR
9364	ClearingVenue	N	—	—	String	If this trade is a cleared SEF trade, LEI of clearing house
9365	TakerLEI	N	—	—	String	LEI of the taker counterparty
9366	MakerLEI	N	—	—	String	Default LEI of the maker counterparty. If the trade is booked with different LEI, please send new maker LEI in execution report. The alternate maker LEI should be pre-configured on Integral's system.
9367	IntentToClear	N	—	—	String	Y/N (Rather than Permitted SEF), Applicable for SEF trades only
9368	EndUserException	N	—	—	String	If the counterparty has invoked clearing exemption, Applicable for SEF trades only
9369	IsBlockTrade	N	—	—	String	Y/N (Y if block trade), Applicable for SEF trades only
9370	ReportingParty	N	—	—	String	Y/N - Is liquidity provider (maker) the reporting party, If US Person, reporting party is mandatory
9374	USINamespace	N	—	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9375	USIFarNamespace	N	—	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
9376	USI	N	—	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	—	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9378	UTIPrefix	N	—	—	String	UTI Namespace for single-leg trades (outrights) and near leg of swaps
9379	UTIFarPrefix	N	—	—	String	UTI Namespace for far leg of swaps. Not applicable to single-leg trades.
9380	UTI	N	—	—	String	UTI value for single-leg trades (outrights) and near leg of swaps
9381	UTIFar	N	—	—	String	UTI value for far leg of swaps. Not applicable to single-leg trades.
9382	UPI	N	—	—	String	UPI value
7556	ExecEndTime	See descr.	—	—	UTCTimestamp	Absolute time in GMT at which the algo should stop executing. If this tag is not specified and `ExecEndPeriod` (#7565) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.
The following shaded rows are key/value pairs specified in the AlgoParameters (#7560) field that represent an algo order's parameters. Each parameter key/value pair is delimited by a space. The tilde character (~) separates the parameter key from the parameter value. For example, the value "SI~01:00:00 SMI~00:30:00 SIR~Y AE~1" indicates "Slice interval = 1 hour, Minimum slice interval = 30 minutes, Slice interval randomized between 30 minutes (SMI) and 1 hour (SI), At expiration fill the remainder at market".						
7560	AlgoParameters	See descr.	Algo	—	MultipleValue String	The algo parameters represented as key/value pairs with the tilde character (~) separating the parameter key from the parameter value and each key/value pair delimited by a space. This field is required when `ExecInst` (#18)=ST.
AE	ActnOrdrExpr	N	Algo	1=Fill @ Market 2=Cancel Order	char	Action to be taken at the expiration of the order. If this tag is not specified, the order is cancelled at expiration.
POT	PegOffsetType	N	Algo	1=Primary 2=Market 3=Mid 4=Transition	char	Indicates that the algo is a pegged order and specifies the peg offset type: ■ Primary: Bid if buying, offer if selling ■ Market: Bid if selling, offer if buying ■ Mid: Midpoint between the bid and the offer ■ Transition: First pegged to primary, then over the offset increment interval (`PII`, `PegOffsetIncrIntrvl`), moves to the market peg in increments by the offset increment (`POI`, `PegOffsetIncr`).

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
PO	PegOffset	N	Algo	—	Price	The amount in pips that is added to the price of the peg. Can be positive or negative.
POI	PegOffsetIncr	N	Algo	—	Price	The fixed amount in pips used to adjust the price offset for transition pegged orders
PII	PegOffsetIncrIntrvl	N	Algo	—	int	The interval at which the offset increment is applied for transition pegged orders. Specified in milliseconds.
PIR	PegOffsetIncrRndmzr	N	Algo	■ Y ■ N	Boolean	Whether or not the offset increment is randomized for transition pegged orders. When the offset increment is randomized, the increment value is a random value between zero and the offset increment (`POI`, `PegOffsetIncr`).
PT	PegTime	N	Algo	—	float	Peg time in seconds. For example, ten and a half seconds is represented as "10.5".
SFOK	SlcFillOrKill	N	Algo	Y=FOK N=IOC (default)	Boolean	Determines how slices are executed. Y=Slices must be filled completely by a single fill from a provider or cancelled. This parameter is used only if the slice size (`SS`, `SlcSize`) is fixed. N=Partial fills allowed.
SI	SlcIntrvl	Y	Algo	—	UTCTimeOnly	Relative time between two slices specified in the format *hh:mm:ss*[*.sss*]. This field is required when `ExecInst` (#18)=ST.
SIR	SlcIntrvlRndmzr	N	Algo	■ Y ■ N	Boolean	Randomization enabled (Y)/disabled (N). If enabled, the order managed system randomly selects a slice interval between `SlicIntrvl` (`SI`) and slice interval and `SlcMinIntrvl` (`SMI`).
SMI	SlcMinIntrvl	Y	Algo	—	UTCTimeOnly	Minimum slice interval specified in the format *hh:mm:ss*[*.sss*]. Applicable if the `SlicIntrvlRndmzr` (`SIR`) parameter is true. This field is required when `ExecInst` (#18) =ST.
SRS	SlcRegularSize	N	Algo	—	Qty	If this parameter is specified, the slice size is rounded to a multiple of this value. If not specified, the behavior defaults to the system configuration. Contact your Integral Solutions Manager to configure the default slice size.
SS	SlcSize	N	Algo	—	Qty	Size of each slice. If the slice size is not specified, the size is chosen based on the `TOBPercent` (`TOBP`) parameter.
SSR	SlcSizeRndmzr	N	Algo	■ Y ■ N	Boolean	Randomization enabled (Y)/disabled (N) flag. If enabled, the order management system randomizes the slice size between `SlcRegularSize` (`SRS`) and `SlcSize` (`SS`) or, if `TOBPercent` (`TOBP`) is specified, between the calculated top-of-book amount and `SlcSize` (`SS`).

Table 4-1 *Order Cancel/Replace Request Message Fields (continued)*

Tag	Field Name	Req'd	Req'd	Value	FIX Format	Description
TOBP	TOBPercent	N	Algo	—	float	Determines how the slice size varies, calculated as a percentage of the top-of-book (TOB) size. For example, a value of 0.5 indicates 50% of TOB size. A value of 1.2 indicates 120% of TOB size. If `SlcSizeRndmzr` (`SSR`) is true, then the slice size varies between `SlcRegularSize` (`SRS`) and the size calculated with the value of this parameter. If this parameter is not specified, the slice size is fixed.
End of field parameters: AlgoParameters (#7560)						
7561	AlgoName	N	Algo	—	String	This field is used with algo orders (`ExecInst` (#18)=ST). A free-format string provided to record the algo employed by the order. The value of this field is not validated and has no effect on algo functionality.
7564	ExecEffPeriod		Algo	—	UTCTimeOnly	Relative time at which the algo should start execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `EffectiveTime` (#168) is not specified, the algo starts executing immediately.
7565	ExecEndPeriod		Algo		UTCTimeOnly	Relative time at which the algo should stop execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `ExecEndTime` (#7556) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.
7700	REXParameters	See descr.	—	TIV~<integer value in seconds>	String	Required for RiskNet directed orders: TIV (Time in Venue), the time in the trading venue specified in seconds. Should be less than `TimeInForce` (#59).
The following shaded rows are a repeating group of fields that represent a user. The required fields must be included as a group for each user. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of users.						
453	NoPartyIDs	N	—	1	NumInGroup	The number of groups of `PartyID` (#448), `PartyIDSource` (#447) and `PartyRole` (#452) fields that represent the end-user ID who submits the order.
448	PartyID	N	—	—	String	The ID of the user who submits the order. See "Users" on page 57 and "Summaries of ID Values" on page 58.
447	PartyIDSource	See descr.	—	D=Proprietary/Custom code	char	Identifies class or source of the `PartyID` (#448) value. Required if `PartyID` is specified.
452	PartyRole	N	—	3=Client ID	int	Identifies the type or role of the `PartyID` (#448) specified.
End of repeating group: NoPartyIDs (#453)						

4.3.5 Order Cancel Reject

(FX Grid to Client)

FX Grid sends an Order Cancel Reject message to the client in response to an unsuccessful Order Cancel Request or Order Cancel/Replace Request message.

***Table 4-1** Order Cancel Reject Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	9	String	9=Order Cancel Reject
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	See descr.	—	String	■ The originating legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer
115	OnBehalfOfCompID	Y	—	String	The liquidity provider organization's ID. See "Summaries of ID Values" on page 58.
128	DeliverToCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
11	ClOrdID	Y	—	String	A session-scoped unique ID assigned by the FIX client for this cancel request being rejected
37	OrderID	Y	—	String	The order ID assigned by the FIX server that could not be canceled or replaced. If the order ID cannot be determined or if the order is not active, the string "NONE" is specified.
39	OrdStatus	Y	—		Order Status value after the cancel reject is applied. Possible Values: ■ A=Pending New: only used in the RFS workflow. See "Request for Stream (RFS) Workflow" on page 17. ■ 0=New ■ 1=Partially filled ■ 2=Filled ■ 8=Rejected ■ C=Expired ■ 4=Canceled ■ 5=Replaced
41	OrigClOrdID	Y	—	String	The `ClOrdID` (#11) of the order that the FIX client wants to cancel
58	Text	N	—	String	Free format text string
102	CxlRejReason	N	—	int	Reasons for cancel rejection: ■ 0=Too late to cancel ■ 1=Unknown order or other reason ■ 3=Order already in Pending Cancel or Pending Replace status ■ 6=Duplicate `ClOrdID` (#11) received
434	CxlRejResponseTo	Y	1	int	The type of request that has been rejected: ■ 1=Order Cancel Request ■ 2=Order Cancel/Replace Request

4.3.6 Order Mass Cancel Request

(Client to FX Grid)

The client sends this message to FX Grid to request that all active orders be canceled.

Groups of associated, repeating fields are indicated by shading in the table below.

Table 4-1 *Order Mass Cancel Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	q	String	q=Order Mass Cancel Request
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider. See "Message Sender and Target" on page 54 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
11	ClOrdID	Y	—	String	A unique ID for the mass cancel request assigned by the FIX client
37	OrderID	Y	—	String	Unique ID generated by the FIX server
60	TransactTime	Y	—	UTCTimestamp	Time this cancel request was initiated/released by the FIX client in the format *YYYYMMDD-hh:mm:ss.sss*.
460	Product	Y	4	int	The asset class. The value is always 4=CURRENCY.
530	MassCancelRequestType	Y	7	char	Supported values are as follows: 7=Cancel all orders
The following shaded rows are a repeating group of fields that represent a user. The required fields must be included as a group for each user. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of users.					
453	NoPartyIDs	N	1	NumInGroup	The number of groups of `PartyID` (#448), `PartyIDSource` (#447) and `PartyRole` (#452) fields that represent the end-user ID who submitted the mass cancel request
448	PartyID	N	—	String	The ID of the user who submitted the mass cancel request. See "Users" on page 57 and "Summaries of ID Values" on page 58.

Table 4-1 *Order Mass Cancel Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
447	PartyIDSource	See descr.	D=Proprietary/Custom code	char	Identifies class or source of the `PartyID` (#448) value. Required if `PartyID` (#448) is specified.
452	PartyRole	N	3=Client ID	int	Identifies the type or role of the `PartyID` (#448) specified.
End of repeating group: NoPartyIDs (#453)					

4.3.7 Order Mass Cancel Report

(FX Grid to Client)

FX Grid sends this message to the client in response to an Order Mass Cancel Request message. Groups of associated, repeating fields are indicated by shading in the table below.

Table 4-1 *Order Mass Cancel Report Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	r	String	r=Order Mass Cancel Report
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	See descr.	—	String	■ The originating legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer
115	OnBehalfOfCompID	Y	—	String	The liquidity provider organization's ID. See "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	Y	—	String	Only cancel orders for the specified ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer
128	DeliverToCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
11	ClOrdID	Y	—	String	A unique ID for this mass cancel request assigned by the FIX client
37	OrderID	Y	—	String	The order ID assigned by the FIX server that could not be canceled or replaced. If all orders were cancelled successfully, the value is "NONE".
58	Text	N	—	String	Free format text string

***Table 4-1** Order Mass Cancel Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
60	TransactTime	Y	—	UTCTimestamp	Time this cancel request was initiated/released by the FIX client in the format *YYYYMMDD-hh:mm:ss.sss*.
530	MassCancelRequestType	Y	—	char	Supported values are as follows: 7=Cancel all orders
531	MassCancelResponse	Y	—	char	Specifies action taken by the FIX server in response to an Order Mass Cancel Request. Possible values: ■ 0=Cancel Request Rejected ■ 7=Cancel all orders
532	MassCancelRejectReason	N	—	char	Reason that Order Mass Cancel Request was rejected. Possible values: ■ 0=Mass Cancel Not Supported ■ 99=Other
533	TotalAffectedOrders	N	—	Int	Total number of orders affected by the request
The following shaded rows are a repeating group of fields that represent the orders affected by the cancel request. The value of the NoAffectedOrders (#534) field indicates the number of groups and thus the number of orders.					
534	NoAffectedOrders	N	—	int	Total number of `OrigClOrdID` (#41) fields of orders affected by the mass cancel request. This number indicates the number of `OrigClOrdID` (#41) fields included in the message.
41	OrigClOrdID	See descr.	—	String	An order ID affected by the mass cancel request. Required if the value of the `NoAffectedOrders` (#534) field is greater than zero.
End of repeating group: NoAffectedOrders (#534)					

4.3.8 Order Status Request

(Client to FX Grid)

The client sends this message to FX Grid to request an execution report be sent to the client with the order's current status.

***Table 4-1** Order Status Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	H	String	H=Order status request
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.

Table 4-1 *Order Status Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
115	OnBehalfOfCompID	See descr.	—	String	The ID of the customer trader who submitted the message if you are trading with liquidity providers on behalf of the customer. Otherwise, this field is not included. See "Users" on page 57 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of your customer's legal entity if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
448	PartyID	Y	—	String	The ID of your organization's user who submitted the message. See "Users" on page 57 and "Summaries of ID Values" on page 58.
11	ClOrdID	Y	—	String	The unique ID assigned by the client to the order, the `ClOrdID` (#11) of the originating New Order – Single message
37	OrderID	N	—	String	The order's ID as assigned by the FIX server. If this field is included, the `ClOrdID` (#11) field is ignored.
54	Side	N	1=Buy (Bid) 2=Sell (Offer)	char	The side of the order from the FIX client's perspective
55	Symbol	N	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")
460	Product	N	4	int	The asset class. The value is always 4=CURRENCY.

4.3.9 Order Mass Status Request

(Client to FX Grid)

This message is sent by the client to request one Execution Report message for each order on the server.

Groups of associated, repeating fields are indicated by shading in the table below.

Table 4-1 *Order Mass Status Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	AF	String	H=Order status request
49	SenderCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	■ Your legal entity's ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58 for more details. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer. The message is rejected if this field is set.

Table 4-1 *Order Mass Status Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
56	TargetCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
115	OnBehalfOfCompID	See descr.	—	String	The customer organization's ID if you are dealing with the liquidity provider as a facilitator on behalf of the customer. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
116	OnBehalfOfSubID	See descr.	—	String	The ID of your customer's trading party if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
128	DeliverToCompID	Y	—	String	The liquidity provider organization's ID. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
584	MassStatusReqID	Y	—	String	The unique ID of the request assigned by the client
585	MassStatusReqType	Y	—	int	The following values are supported: 7=Status for all orders. The query can be further narrowed down by including a customer entity ID in `OnBehalfOfCompID` (#115). 8=Status for orders for a user. The following fields must be included to specify the user whose order statuses are requested: ■ `NoPartyIDs` (#453) ■ `PartyID` (#448,) ■ `PartyIDSource` (#447) ■ `PartyRole` (#452)
The following shaded rows are a repeating group of fields that represent a user. The required fields must be included as a group for each user. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of users.					
453	NoPartyIDs	N	1	NumInGroup	The number of groups of `PartyID` (#448), `PartyIDSource` (#447) and `PartyRole` (#452) fields that represent the end-user ID who submitted the request
448	PartyID	N	—	String	The ID of the user who submitted the request. See "Users" on page 57 and "Summaries of ID Values" on page 58.
447	PartyIDSource	See descr.	D=Proprietary/Custom code	char	Identifies class or source of the `PartyID` (#448) value. Required if `PartyID` is specified.
452	PartyRole	N	3=Client ID	int	Identifies the type or role of the `PartyID` (#448) specified.
End of repeating group: NoPartyIDs (#453)					

4.3.10 Execution Report

(FX Grid to Client)

This message is sent by FX Grid in the following events:

- Order filled (full or partial)

- Order submission failed
- Order status request received from the client
- Order mass status request received from the client

The combination of the `OrdStatus` (#39) and `ExecType` (#150) fields indicate the current state of the order.

If order submission fails for a liquidity provider, you can retry execution with that provider when you receive the next rate update from the provider.

Table 4-1 *Execution Report Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	8	String	8=Execution Report
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
50	SenderSubID	See descr.	—	String	The liquidity provider organization's legal entity for provider-originated messages, such as trade verification. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
57	TargetSubID	See descr.	—	String	■ The originating legal entity or trading party ID if you are dealing directly with the liquidity provider. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58. ■ Not included in the message if you are dealing with the liquidity provider as a facilitator on behalf of the customer
115	OnBehalfOfCompID	Y	—	String	The liquidity provider organization's ID
116	OnBehalfOfSubID	N	—	String	When the stream LP trading workflow is used, the organization ID of the stream LP is sent in this field.
128	DeliverToCompID	See descr.	—	String	The customer organization's ID if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included. See "Business Sender and Target" on page 55 and "Summaries of ID Values" on page 58.
129	DeliverToSubID	See descr.	—	String	The ID of your customer's trading party if you are dealing with the liquidity provider as a facilitator on behalf of the customer. Otherwise, this field is not included in the message. See "Legal Entities and Trading Parties" on page 57 and "Summaries of ID Values" on page 58.
6	AvgPx	Y	—	Price	The average execution price of fills on the order. Compare with the `LastPx` (#31) field. The precision of this float value must be agreed upon by both the sending and the receiving parties.
11	ClOrdID	Y	—	String	A unique order ID assigned by the client. It is the same value as sent by the client in the New Order - Single message.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
14	CumQty	Y	—	Qty	FX spot, outright: Total filled amount. If the `ExecType` (#150) field is "2", the value of this field is the same as the `OrderQty` (#38) field. If the `ExecType` (#150) field is "8", the value of this field is "0" (zero). FX swap: Near-leg total filled amount. If the `ExecType` (#150) field is "2", the value of this field is the same as the `OrderQty` (#38) field. If the `ExecType` (#150) field is "8", the value of this field is "0" (zero).
15	Currency	Y	—	Currency	The dealt currency. This may be the base or term currency of a currency pair.
17	ExecID	Y	—	String	Unique ID for each order execution report. If the value of the `ExecType` (#150) field is "F" (Trade), the value of this tag is the unique FX Grid deal ID. If the value of the `ExecType` field is "I" (Order Status), the value of this tag is "0" (zero).
18	ExecInst	Y	B=OK to cross (execute at market) B not specified=No cross P=Best price P not specified=Price at depth 9=Bid trigger 0=Offer trigger M=Mid trigger R=At rate ST=Algo W=VWAP VN=Venue	MultipleValueString	How your order is executed. As applicable to the order type, you can specify more than one value by separating each value with a space. For details about how this field applies to the various order types, See "Order Execution" on page 30. Required for RiskNet: "VN" and "B" both to indicate that the order is directed to a specific venue (as specified by `DeliverToCompID` (#128)=Trading venue ID) and that the order can executed at market if the order is not completely filled within the time specified by `REXParameters` (#7700)=TIV~<number of seconds>.
20	ExecTransType	N	0=New	char	For batch trades only: Identifies transaction type
31	LastPx	N	—	Price	FX spot, outright: Price at which the current or last fill was made. FX swap: Near-leg price at which the current or last fill was made. Compare with the `AvgPx` (#6) field.
32	LastQty	N	—	Qty	Amount of the current or last fill.
37	OrderID	Y	—	String	The unique order ID assigned by the FIX server.
38	OrderQty	Y	—	Qty	FX spot, outright: The "Total Intended Order Quantity" (including the amount already filled for this chain of orders) expressed in the dealt currency. FX swap: Near-leg "Total Intended Order Quantity" (including the amount already filled for this chain of orders) expressed in the dealt currency.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
39	OrdStatus	Y	—	char	Describes the current state of the order. Valid values: ■ A=Pending New: only used in the RFS workflow. See "Request for Stream (RFS) Workflow" on page 17. ■ 0=New (outstanding) ■ D=Accepted for bidding ■ 1=Partial Fill (after order matching) ■ 2=Filled (after order matching) ■ 7=Stopped (for algo orders with delayed start and suspended orders) ■ 8=Rejected ■ C=Expired ■ 4=Canceled ■ 5=Replaced ■ 6=PendingCancel ■ E=PendingReplace
40	OrdType	N	■ 1=Market or Market Range ■ 2=Limit (orders) ■ 3=Stop ■ 4=Stop Limit ■ D=Previously quoted (ESP, RFS, or NDF)	char	Other order types are not currently supported. See "Supported Order Types" on page 27. If the order type is 1, then the value of the `PegOffsetValue` (#211) field determines whether the order is a market order or a market range order. See "PegOffsetValue" on page 97.
41	OrigClOrdID	N	—	String	The unique ID of the replaced order assigned by the client. Required if the submitted order's `ExecType` (#150) field value is PendingReplace or Replace.
44	Price	Y	—	Qty	This should be the same value as the one received from the associated New Order - Single message. Do not use this value to determine the price of fills. Instead, use `LastPx` (#31) for an individual fill or `AvgPx` (#6) for the average price of all fills on the order. FX spot, outright: The execution price for limit and previously quoted orders. The precision of this float value must be agreed upon by both the sending and the receiving parties. FX swap: Near-leg execution price.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
54	Side	Y	1=Buy (Bid) 2=Sell (Offer)	char	The side of the order from the customer's perspective. For FX swap orders, this is the side of the far leg. This should be the same value as was sent on the New Order – Single message. If you are dealing directly with the liquidity provider, the customer is defined by the `TargetCompID` (#56) and `TargetSubID` (#57) fields. If you are dealing with the liquidity provider as a facilitator on behalf of the customer, the customer is defined by the `DeliverToCompID` (#128) and `DeliverToSubID` (#129) fields.
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD") This should be the same value as the one received from the associated New Order - Single message.
58	Text	N	—	String	Free format text explaining the reason for the failure if the Order Execution Report message is sent because of order rejection (the `ExecType` (#150) field="8"). If the `Text` (#58) field includes the text "INTERNAL_SERVER_ERROR", this indicates a serious error. Stop trading immediately and contact Business Support to resolve the issue.
59	TimeInForce	Y	■ 0 (zero)=Day ■ 1=GTC ■ 3=IOC ■ 4=FOK ■ 6=GTT	char	See "Order Expiry" on page 31 for a description of order expiry types.
60	TransactTime	Y	—	UTCTimestamp	If the `ExecType` (#150) field is "2", the value of this field is the business execution timestamp (the date and time of the order match). If the `ExecType` (#150) field is "8", the value of this field is the time the order failed. The format is *YYYYMMDD-hh:mm:ss.sss*
64	FutSettDate	N	—	LocalMktDate	The field is specified in the format *YYYYMMDD*. ■ FX spot: Spot date ■ FX outright: Outright value date ■ FX spot-forward swap: Spot date ■ FX forward-forward swap: Near value date
75	TradeDate	N	—	LocalMktDate	The trade date in the format *YYYYMMDD*

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
103	OrdRejReason	N	—	int	The reason for the order submission failure resulting in a reject message. See "Order Submission Failure" on page 19 for more information. ■ 0=Broker/Exchange option ■ 1=Unknown symbol ■ 2=Exchange closed ■ 3=Order exceeds limit ■ 4=Too late to Enter ■ 5=Unknown Order ■ 6=Duplicate Order ■ 7=Duplicate of a verbally communicated order ■ 8=Stale Order ■ 9=Trade Along required (FIX4.3 and later) ■ 10=Invalid Investor ID (FIX4.3 and later) ■ 11=Unsupported order characteristic (FIX4.3 and later) ■ 12=Surveillance Option (FIX4.3 and later) ■ 13=Incorrect quantity (FIX 4.4, used when an order fails due to pre-allocation information errors) ■ 14=Incorrect allocated quantity (FIX 4.4, used when an order fails due to pre-allocation information errors) ■ 15=Unknown account(s) (FIX 4.4, used when an order fails due to pre-allocation information errors) ■ 99=Other (FIX 4.4)
110	MinQty	N	—	Qty	Minimum amount of the order that was requested to be executed. See "Minimum Order Size" on page 33. This field is ignored if `TimeInForce` (#59) is 4 (FOK).
119	SettlCurrAmt	N	—	Amt	FX spot, outright: Settled amount in terms of settlement currency specified by the `SettlCurrency` (#120) field. FX swap: Near-leg settled amount in terms of settlement currency specified by the `SettlCurrency` (#120) field.
120	SettlCurrency	N	—	Currency	Settled currency
126	ExpireTime	N	—	UTCTimestamp	The relative expiration time of the order in the format *YYYYMMDD-hh:mm:ss.sss*. Required when the `TimeInForce` (#59) value is 6 (GTD). The date portion of the value is ignored. The time portion specifies the amount of time before the order expires. For example, for an order that expires in ten seconds, the value could be 00000000-00:00:10.000.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
150	ExecType	Y	—	char	Describes the type of execution report. Valid values: ■ 0=New (optional, see "Order Workflow" on page 13) ■ 1=Partial Fill ■ 2=Fill ■ 4=Canceled ■ 5=Replace ■ 6=PendingCancel ■ 8=Rejected ■ A=Pending New (optional, see "Order Workflow" on page 13): only used in the RFS workflow. See "Request for Stream (RFS) Workflow" on page 17. ■ C=Expired ■ D=Restated ■ E=PendingReplace ■ F=Trade (Partial Fill or Fill) ■ I=Order Status
151	LeavesQty	Y	—	Qty	FX spot, outright: Open amount This value cannot be negative. For the following deterministic order states, a value of "0" (zero) indicates that the order has been fully filled and a value greater than zero indicates that the order is not fully filled: ■ `ExecType` (#150)=4 (Canceled) ■ `ExecType` (#150)=5 (Replace) ■ `ExecType` (#150)=6 (PendingCancel) ■ `ExecType` (#150)=8 (Rejected) ■ `ExecType` (#150)=C (Expired) ■ `ExecType` (#150)=D (Restated) ■ `ExecType` (#150)=F (Trade, Partial Fill or Fill) ■ `ExecType` (#150)=I (Order Status) For the following nondeterministic states, the value of this field should be disregarded: ■ `ExecType` (#150)=0 (New) ■ `ExecType` (#150)=A (Pending new) ■ `ExecType` (#150)=E (PendingReplace)
167	SecurityType	Y	FOR	String	FOR=Foreign Exchange Contract
168	EffectiveTime	N	—	UTCTimestamp	The order submission time in GMT in the format *YYYYMMDD-hh:mm:ss* or *YYYYMMDD-hh:mm:ss.sss*. For algo orders, the value of this field indicates, he absolute time in GMT at which the algo should start execution. If this field is not specified and `ExecEffPeriod` (#7564) is not specified, the algo starts execution immediately. If this field is specified and the time value is in the past, the algo starts execution immediately.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
192	OrderQty2	N	—	Qty	FX spot, outright: Not applicable. FX swap: Far-leg "Total Intended Order Quantity" (including the amount already filled for this chain of orders) expressed in the dealt currency.
193	FutSettDate2	N	■ Tenor: See "Supported Tenors" on page 26. ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either standard tenor symbol or a broken date in the format *YYYYMMDD*.
194	LastSpotRate	N	—	Price	FX spot: Not applicable. FX outright: Spot rate FX swap: Near-leg spot rate
195	LastForwardPoints	N	—	PriceOffset	FX spot: Not applicable. FX outright: Forward points FX swap: Near-leg forward points
210	MaxShow	N	■ 0 (zero) ■ Equal to `OrderQty` (#38) ■ Less than `OrderQty` (#38)	Qty	The amount of the order visible to other market participants based on the value of the field: ■ Hidden: If the value is 0 (zero), empty, or not included, the order is a hidden order that is not visible to other customers. ■ Display: If the value is equal to the value of the `OrderQty` (#38) field, the order is a display order that is completely visible by other customers. The full order amount is disclosed. ■ Iceberg: If the value of is less than the value of the `OrderQty` (#38) field, the order is an iceberg order that is visible to other customers. Only a fraction of the actual order amount is displayed (the value of this field).
460	Product	See descr.	4	int	The asset class. The value is always 4=CURRENCY. This field is not required for batch trades.
541	MaturityDate	See descr.	N	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64). NDF swaps: The near-leg fixing date
8541	FarMaturityDate	See descr.	N	LocalMktDate	For NDF swaps only: The far-leg fixing date
The following shaded rows are a repeating group of fields that represent a trade leg in the batch of trades. The value of the NoLegs (#555) field indicates the number of groups and thus the number of legs.					
555	NoLegs	Y	—	NumInGroup	Repeating group for multi-leg trade details
600	LegSymbol	Y	—	String	The leg's currency pair

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
624	LegSide	Y	—	char	The side of the leg: Empty=2-way 1=Buy 2=Sell
556	LegCurrency	Y	—	Currency	The dealt currency of the leg
654	LegRefID	Y	—	String	The unique ID of the trade represented by the leg in the batch. For multi-leg trades, such as swap trades, this ID should be the same for all legs of the trade.
566	LegPrice	N	—	Price	All-in rate of the executed trade
588	LegSettDate	Y	—	LocalMktDate	Value date of the leg
1893	LegExecID	N	—	String	The ID of the executed trade
7632	MidPxLeg	N	—	Price	For SEF trades: The mid rate
End of repeating group: NoLegs (#555)					
583	ClOrdLinkID	N	—	String	The IDs of the orders to which this order is linked. This field is only populated in a New Order – Single message when submitting a component order of an OCO or OUO order chain. The value should be the `ClOrdID` (#11) values of the other component orders. See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.
584	MassStatusReqID	N	—	String	Uniquely identifies a specific Order Mass Status Request message. Required if the Execution Report is a response to an Order Mass Status Request. The `ExecType` (#150) is "I=Order Status" for this Execution Report.
631	MidPx	N	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
636	WorkingIndicator	N	■ Y ■ N	Boolean	For stop and stop limit orders a value of "Y" indicates that the stop has been triggered and the resulting market, market range, or limit order is working. If "N" or empty, or if the field is not included in the message, the stop has not been triggered. See "Order Workflow" on page 13 and "Supported Order Types" on page 27.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
640	Price2	N	—	Qty	This should be the same value as the one received from the associated New Order - Single message. FX spot, outright: Not applicable. FX swap: Far-leg execution price.
641	LastForwardPoints2	N	—	PriceOffset	FX spot: Not applicable. FX outright: Not applicable. FX swap: Far-leg forward points
790	OrdStatusReqID	N	—	String	Uniquely identifies a specific Order Status Request message. Required if the Execution Report is a response to an Order Status Request. The `ExecType` (#150) is "I=Order Status" for this Execution Report.
1385	ContingencyType	N	See descr.	int	Defines the type of contingency. 1=One Cancels the Other (OCO) 3=One Updates the Other (OUO) - Absolute Quantity Reduction 4=One Updates the Other (OUO) - Proportional Quantity Reduction See "One-Cancels-the-Other (OCO) Orders" on page 34 and "One-Updates-the-Other (OUO) Orders" on page 35.
7541	LastPx2	N	—	Price	FX spot, outright: Not applicable. FX swap: Far-leg price at which the current or last fill was made.
7542	LastSpotRate2	N	—	Price	FX spot, outright: Not applicable. FX swap: Far-leg spot rate. Only sent under very specific conditions for RFS order rejections.
7543	LeavesQty2	N	—	Qty	FX spot, outright: Not applicable. FX swap: Far-leg open amount
7544	CumQty2	N	—	Qty	FX spot, outright: Not applicable. FX swap: Far-leg total filled amount. If the `ExecType` (#150) field is "2", the value of this field is the same as the `OrderQty` (#38) field. If the `ExecType` (#150) field is "8", the value of this field is "0" (zero).
7545	SettlCurrAmt2	N	—	Amt	FX spot, outright: Settled amount in terms of settlement currency specified by the `SettlCurrency` (#120) field. FX swap: Far-leg settled amount in terms of settlement currency specified by the `SettlCurrency` (#120) field.
9366	MakerLEI	N	—	String	Legal Entity Identifier (LEI) of the maker of the trade. If there are multiple LEIs with which the maker can book the trade, please specify the final LEI with which the trade was booked.
9374	USINamespace	N	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
9375	USIFarNamespace	N	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9376	USI	N	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9378	UTIPrefix	N	—	String	UTI Namespace for single-leg trades (outrights) and near leg of swaps
9379	UTIFarPrefix	N	—	String	UTI Namespace for far leg of swaps. Not applicable to single-leg trades.
9380	UTI	N	—	String	UTI value for single-leg trades (outrights) and near leg of swaps
9381	UTIFar	N	—	String	UTI value for far leg of swaps. Not applicable to single-leg trades.
7556	ExecEndTime	See descr.	—	UTCTimestamp	Absolute time in GMT at which the algo should stop executing. If this tag is not specified and `ExecEndPeriod` (#7565) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.
The following shaded rows are key/value pairs specified in the AlgoParameters (#7560) field that represent an algo order's parameters. Each parameter key/value pair is delimited by a space. The tilde character (~) separates the parameter key from the parameter value. For example, the value "SI~01:00:00 SMI~00:30:00 SIR~Y AE~1" indicates "Slice interval = 1 hour, Minimum slice interval = 30 minutes, Slice interval randomized between 30 minutes (SMI) and 1 hour (SI), At expiration fill the remainder at market".					
7560	AlgoParameters	See descr.	—	MultipleValueString	The algo parameters represented as key/value pairs with the tilde character (~) separating the parameter key from the parameter value and each key/value pair delimited by a space. This field is required when `ExecInst` (#18)=ST.
AE	ActnOrdrExpr	N	1=Fill @ Market 2=Cancel Order	char	Action to be taken at the expiration of the order. If this tag is not specified, the order is cancelled at expiration.
POT	PegOffsetType	N	1=Primary 2=Market 3=Mid 4=Transition	char	Indicates that the algo is a pegged order and specifies the peg offset type: ■ Primary: Bid if buying, offer if selling ■ Market: Bid if selling, offer if buying ■ Mid: Midpoint between the bid and the offer ■ Transition: First pegged to primary, then over the offset increment interval (`PII`, `PegOffsetIncrIntrvl`), moves to the market peg in increments by the offset increment (`POI`, `PegOffsetIncr`).

Table 4-1 *Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
PO	PegOffset	N	—	Price	The amount in pips that is added to the price of the peg. Can be positive or negative.
POI	PegOffsetIncr	N	—	Price	The fixed amount in pips used to adjust the price offset for transition pegged orders
PII	PegOffsetIncrIntrvl	N	—	int	The interval at which the offset increment is applied for transition pegged orders. Specified in milliseconds.
PIR	PegOffsetIncrRndmzr	N	■ Y ■ N	Boolean	Whether or not the offset increment is randomized for transition pegged orders. When the offset increment is randomized, the increment value is a random value between zero and the offset increment (`POI`, `PegOffsetIncr`).
PT	PegTime	N	—	float	Peg time in seconds. For example, ten and a half seconds is represented as "10.5".
SFOK	SlcFillOrKill	N	Y=FOK N=IOC (default)	Boolean	Determines how slices are executed. Y=Slices must be filled completely by a single fill from a provider or cancelled. This parameter is used only if the slice size (`SS`, `SlcSize`) is fixed. N=Partial fills allowed.
SI	SlcIntrvl	Y	—	UTCTimeOnly	Relative time between two slices specified in the format *hh:mm:ss*[*.sss*]. This field is required when `ExecInst` (#18)=ST.
SIR	SlcIntrvlRndmzr	N	■ Y ■ N	Boolean	Randomization enabled (Y)/disabled (N). If enabled, the order managed system randomly selects a slice interval between `SlicIntrvl` (`SI`) and slice interval and `SlcMinIntrvl` (`SMI`).
SMI	SlcMinIntrvl	Y	—	UTCTimeOnly	Minimum slice interval specified in the format *hh:mm:ss* [*.sss*]. Applicable if the `SlicIntrvlRndmzr` (`SIR`) parameter is true. This field is required when `ExecInst` (#18)=ST.
SRS	SlcRegularSize	N	—	Qty	If this parameter is specified, the slice size is rounded to a multiple of this value. If not specified, the behavior defaults to the system configuration. Contact your Integral Solutions Manager to configure the default slice size.
SS	SlcSize	N	—	Qty	Size of each slice. If the slice size is not specified, the size is chosen based on the `TOBPercent` (`TOBP`) parameter.
SSR	SlcSizeRndmzr	N	■ Y ■ N	Boolean	Randomization enabled (Y)/disabled (N) flag. If enabled, the order management system randomizes the slice size between `SlcRegularSize` (SRS) and `SlcSize` (`SS`) or, if `TOBPercent` (`TOBP`) is specified, between the calculated top-of-book amount and `SlcSize` (SS).

***Table 4-1** Execution Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
TOBP	TOBPercent	N	—	float	Determines how the slice size varies, calculated as a percentage of the top-of-book (TOB) size. For example, a value of 0.5 indicates 50% of TOB size. A value of 1.2 indicates 120% of TOB size. If `SlcSizeRndmzr` (`SSR`) is true, then the slice size varies between `SlcRegularSize` (`SRS`) and the size calculated with the value of this parameter. If this parameter is not specified, the slice size is fixed.
End of field parameters: AlgoParameters (#7560)					
7561	AlgoName	N	—	String	This field is used with algo orders (`ExecInst` (#18) =ST). A free-format string provided to record the algo employed by the order. The value of this field is not validated and has no effect on algo functionality.
7564	ExecEffPeriod		—	UTCTimeOnly	Relative time at which the algo should start execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `EffectiveTime` (#168) is not specified, the algo starts executing immediately.
7565	ExecEndPeriod			UTCTimeOnly	Relative time at which the algo should stop execution specified in the format *hh:mm:ss*[*.sss*]. If this tag is not specified and `ExecEndTime` (#7556) is not specified, the order expires based on the value of the `TimeInForce` (#59) field.

4.3.11 Business Message Reject

The Business Message Reject message is sent by both the server and the client.

FX Grid sends a Business Message Reject message in the following scenarios:

- In the order workflow to reject an order status request if the order does not exist (`RefMsgType` (#372)=`H`). See "Order Workflow" on page 13.
- In the batch trade workflow when a client's request to withdraw an RFS stream is not successful. See "Batch Trade Workflow" on page 18.
- In the post-trade workflow to reject a trade status request. See "Trade Status Query" on page 23.

***Table 4-1** Business Message Reject Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	j	String	j=Business Message Reject
49	SenderCompID	Y	—	String	The FX Inside server's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
56	TargetCompID	Y	—	String	Your organization's ID. See "Message Sender and Target" on page 54 and "Summaries of ID Values" on page 58.
58	Text	N	—	String	Free-format text string

Table 4-1 *Business Message Reject Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
372	RefMsgType	Y	—	String	The `MsgType` (#35) of the rejected message: H=Order Status Request (from server to reject an order status request if the order does not exist) Z=Quote Cancel (from client to cancel a Quote Request)
379	BusinessRejectRefID	N	—	String	The ID of the rejected message
380	BusinessRejectReason	Y (from server) N (from client)	See descr.	int	Applies only to the order status reject from the server. Code to identify reason for a Business Message Reject message. Currently, two values are supported: 0=Other 1=Unknown ID

4.4 Position Management Messages

The messages in this section are used for querying position status.

You can request a position snapshot or subscribe to a stream of position data that updates dynamically as trades are done.

4.4.1 Request for Positions

(Client to FX Grid)

The Request for Positions message is sent by the client to request position status and to subscribe to position updates.

Table 4-1 *Request for Positions Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y		String	AN=Request for Positions Message
60	TransactTime	Y	—	UTCTimestamp	The time that the position request was initiated in GMT in the format *YYYYMMDD-hh:mm:ss.sss*.
263	SubscriptionRequestType	Y	0=Snapshot 1=Snapshot + Updates (Subscribe) 2=Disable previous Snapshot + Update Request (Unsubscribe)	char	Indicates type of response requested. Snapshot requests a single message response. A subscribe request asks for position updates as trades are executed. Unsubscribe cancels any future update messages.
710	PosReqID	Y			Unique identifier for the Request for Positions as assigned by the trader
55	Symbol	Y	—	String	The symbol for the base and variable currencies of the currency pair in the following format: *baseCCY/variableCCY* (for example, "EUR/USD")

Table 4-1 *Request for Positions Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
724	PosReqType	Y	0=Open positions 1=Settled positions 2=Both open and settled positions	int	Type of position requested
The following shaded rows are a group of fields that represent parties on positions. The value of the NoPartyIDs (#453) field is always 1 (one). The required fields must be included as a group for each party.					
453	NoPartyIDs	Y	1 (one)	NumInGroup	The number of groups
448	PartyID	Y	ALL	String	The ID of the legal entity for which positions are requested. The value of this field is always "ALL". If `PartyRole` (#452) is not specified and no `NoPartySubIDs` (#802) groups are specified, then all positions for all counterparties are requested.
447	PartyIDSource	Y	D=Proprietary/Custom code	char	The class or source of the `PartyID` (#448) value.
452	PartyRole	See descr.	3=Client ID	Int	To requests all positions for all counterparties, this field is not required. For requests for positions with specific counterparties, this field must be specified.
The following shaded rows are a repeating group of fields that represent parties on positions. The value of the NoPartySubIDs (#802) field indicates the number of groups and thus the number of parties requested. These fields are required for requests for positions with specific counterparties.					
802	NoPartySubIDs	N	—	NumInGroup	The number of parties specified
523	PartySubID	N	—	String	The ID of the legal entity for which positions are requested.
803	PartySubIDType	N	1=Organization 4001=Counterparty	String	The type of the party specified by the `PartySubID` (#523): 1 (one)=Organization. Required for sales dealer organizations. Specifies the customer organization. 4001=Counterparty
End of repeating group: NoPartySubIDs (#802)					
End of repeating group: NoPartyIDs (#453)					
7603	DateType	See descr.	0=Value date 1=Trade date	int	Describes the dates specified in `StartDate` (#916) and `EndDate` (#917) fields.
916	StartDate	N	—	LocalMktDate	Do not include this field if requesting settled positions. The start date of the date range for open position requests in format *YYYYMMDD*.
917	EndDate	N	—	LocalMktDate	Do not include this field if requesting settled positions. The end date of the date range for open position requests in format *YYYYMMDD*.

4.4.2 Request for Positions Ack

(FX Grid to Client)

The Request for Positions Ack is sent by the server to the client to acknowledge a successful request for positions or reject an unsuccessful request.

Table 4-1 *Request for Positions Ack Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y		String	AO=Request for Positions Ack Message
58	Text	N	—	String	If the request is rejected, this field can contain the rejection reason.
710	PosReqID	N	—	String	The unique ID of the associated Request for Positions message
721	PosMaintRptID	Y	—	String	The `PosMaintRptID` (#721) value from first position report for a successful request or "0" for rejection.
727	TotNumPosReports	N	—	int	The total number of position reports in the initial reply to a successful request. For an unsolicited report sent as part of a subscription the value of this field is "1". For rejected requests, the value is "0".
728	PosReqResult	Y	0=Valid Request 2=No Positions Found 66=Other	int	The result of the positions request
729	PosReqStatus	Y	0=Completed 2=Rejected	int	The status of the positions request
The following shaded rows are a group of fields that represent values from the associated Request for Positions message.					
453	NoPartyIDs	Y	1 (one)	NumInGroup	The number of groups
448	PartyID	Y	ALL	String	The ID of the legal entity for which positions are requested. The value of this field is always "ALL". If `PartyRole` (#452) is not specified and no `NoPartySubIDs` (#802) groups are specified, then all positions for all counterparties are requested.
447	PartyIDSource	Y	D=Proprietary/Custom code	char	The class or source of the `PartyID` (#448) value.
452	PartyRole	See descr.	3=Client ID	Int	To request all positions for all counterparties, this field is not required. For requests for positions with specific counterparties, this field must be specified.
The following shaded rows are a repeating group of fields that represent values from the associated Request for Positions message.					
802	NoPartySubIDs	N	—	NumInGroup	The number of parties specified
523	PartySubID	N	—	String	The ID of the legal entity for which positions are requested.
803	PartySubIDType	N	1=Organization 4001=Counterparty	String	The type of the party specified by the `PartySubID` (#523): 1 (one)=Organization. Required for sales dealer organizations. Specifies the customer organization. 4001=Counterparty
End of repeating group: NoPartySubIDs (#802)					
End of repeating group: NoPartyIDs (#453)					

4.4.3 Positions Report

(FX Grid to Client)

The Positions Report message is sent from the server to the client in response to a successful Request for Positions message. One or more messages can be sent per currency pair depending on

the criteria set by the Request for Positions. This message can be sent solicited as a snapshot or unsolicited as part of a subscription when a trade is executed. Positions Report messages are sent on an order session.

Table 4-1 *Positions Report Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y		String	AP=Positions Report Message
15	Currency	Y	—	Currency	The currency of the position (the base currency of the currency pair)
54	Side		1=Buy 2=Sell	char	Whether the position is long or short from the perspective of the either your organization or your customer's organization when PartySubIDType (#803)=1 (Organization). 1 (Buy) indicates a net long position in the base currency. 0 (Sell) indicates a net short position in the base currency.
60	TransactTime	Y	—	UTCTimestamp	The time that the position report was issued in GMT in the format *YYYYMMDD-hh:mm:ss.sss*.
325	UnsolicitedIndicator		■ Y ■ N	Boolean	Indicates whether or not message is being sent as a result of a subscription request or not. N=Message is being sent as a result of a prior request Y=Message is being sent unsolicited
710	PosReqID	N	—	String	The unique ID of the associated Request for Positions message
721	PosMaintRptID	Y	—	String	The unique ID of this Positions Report message
727	TotNumPosReports	N	—	int	The total number of position reports in the initial reply to a successful request. For an unsolicited report sent as part of a subscription the value of this field is "1".
912	LastRptRequested		■ Y ■ N	Boolean	Indicates whether this message is the last report message in response to the position request. This field is omitted for unsolicited reports that are sent as part of a subscription. Y=Message is the last report N=Message is not the last report
7607	NetAmountBase	Y	—	Qty	The net position in base currency
7608	NetAmountTerm	Y	—	Qty	The net position in term currency
7609	BuyAmountBase	Y	—	Qty	The total amount bought in base currency
7610	SellAmountTerm	Y	—	Qty	The total amount sold in term currency
7611	SellAmountBase	Y	—	Qty	The total amount sold in base currency
7612	BuyAmountTerm	Y	—	Qty	The total amount bought in term currency
7613	AvgBuyRate	Y	—	Price	Weighted average base buy rate
7614	AvgSellRate	Y	—	Price	Weighted average base sell rate
7615	AvgMarketBuyRate	Y	—	Price	Weighted average base buy rate without any spreads
7616	AvgMarketSellRate	Y	—	Price	Weighted average base sell rate with any spreads
7617	MarketPosRate	Y	—	Price	
7618	NetPNL	Y	—	Qty	The PnL if the position is closed at the current market rate

4.5 Post-Trade Messages

The messages in this section are used for STP download of done trades and for querying trade status.

4.5.1 Trade Capture Report Request

(Client to FX Grid)

The client sends this message to the server to query a trade's status by specific trade ID or by a date/time range. Normally, the client should not have to send a Trade Capture Report Request because the server sends a Trade Capture Report asynchronously when a trade is done. This message should be sent only for reconciliation purposes if you think a trade download is missing or is incorrect.

The request can be sent for a specific trade or for all trades in a date range:

- Request for Specific Trade

  To request a specific trade, the value of the `ExecID` (#17) field is set to the trade's ID. The date-range related fields are ignored: `NoDates` (#580), `TradeDate` (#75) and `TransactTime` (#60).

- Request by Date Range

  To request trades within a date/time range, the `ExecID` (#17) field must either not be included in the message or its value must be null. The `NoDates` (#580) must be set to "2", indicating a range with a start date/time and end date/time. For queries by trade date only, the `TradeDate` (#75) field is repeated twice. The first instance is the start date and the second instance is the end date. For queries by trade date and time, the `TransactTime` (#60) field is repeated twice. The first instance is the start date/time and the second instance is the end date/time. If both the `TradeDate` (#75) field and `TransactTime` (#60) are specified on the request, then the `TradeDate` (#75) field is ignored.

***Table 4-1** Trade Capture Report Request Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	AD	String	AD=Trade Capture Report Request
17	ExecID	See descr.	—	String	A specific trading application trade (Transaction) ID. If `ExecID` (#17) is specified on the message, `NoDates` (#580), `TradeDate` (#75) and `TransactTime` (#60) are ignored. If you want to query for trades within a date/time range, leave this field empty.
568	TradeRequestID	Y	—	String	The unique ID of the Trade Capture Report Request message.
569	TradeRequestType	Y	—	int	Valid values: ■ 0=All trades ■ 1=Matched trades matching criteria provided on request ■ 2=Unmatched trades that match criteria ■ 3=Unreported trades that match criteria ■ 4=Advisories that match criteria
The following shaded rows are a repeating group of fields that represent the begin and end of a date range.					

Table 4-1 *Trade Capture Report Request Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
580	NoDates	See descr.	2	Int	Value must be 2 for a date or time range. For queries by trade date only, the `TradeDate` (#75) field is repeated twice. The first instance is the start date and the second instance is the end date. For queries by trade date and time, the `TransactTime` (#60) field is repeated twice. The first instance is the start date/time and the second instance is the end date/time. If both the `TradeDate` (#75) field and `TransactTime` (#60) are specified on the request, then the `TradeDate` (#75) field is ignored.
75	TradeDate	See descr.	—	LocalMktDate	Trade date in format *YYYYMMDD*. Used to request trades by trade date only. The `TradeDate` (#75) field is repeated twice. The first instance is the start date and the second instance is the end date.
60	TransactTime	See descr.	—	UTCTimestamp	The timestamp of the trade in GMT in the format *YYYYMMDD-hh:mm:ss.sss*. Used to request trades by trade date and time. The `TransactTime` (#60) field is repeated twice. The first instance is the start date/time and the second instance is the end date/time. If both the `TradeDate` (#75) field and `TransactTime` (#60) are specified on the request, then the `TradeDate` (#75) field is ignored.
End of repeating group: NoDates (#580)					

4.5.2 Trade Capture Report

(FX Grid to Client)

Message Synchronization

The server sends this message synchronously or asynchronously depending on the workflow involved:

- STP download (asynchronous, triggered by a trade being done, `TradeReportTransType` (#487) =N)
- STP resend (asynchronous, triggered by an administrator, `TradeReportTransType` (#487)=R, `ExecTyp` (#150)=5, `PreviouslyReported` (#570)=Y)
- Trade rebook (asynchronous, triggered by an administrator, `TradeReportTransType` (#487)=R, `ExecTyp` (#150)=5, `PreviouslyReported` (#570)=Y)
- Trade cancel (asynchronous, triggered by cancellation event, `TradeReportTransType` (#487) =C)
- Reverse (FIX 4.4 only, asynchronous, triggered by cancellation/modification event, `TradeReportTransType` (#487)=4)
- Trade status (synchronous, sent in response to a trade status query from the client)

Intrafloor Trades

When intrafloor trades are done in your organization, you may receive STP messages in the following manner depending on how your organization has been configured:

- None: No STP message.
- Taker: One STP message from the taker's perspective (the taker on the intrafloor trade).
- Maker: One STP message from the maker's perspective (the maker on the intrafloor trade).

- Both: Two STP messages; one from the taker's perspective and one from the maker's perspective.

The following table details the ID values for intrafloor trades from the different perspectives.

***Table 4-1** STP Field Values for Intrafloor Trades*

Field	Taker Perspective	Maker Perspective
`SenderSubID` (#50)	Taker's executing legal entity ID (stream LE)	Taker's executing legal entity ID
`TargetSubID` (#57)	Legal entity ID of user who submitted the displayed order	Legal entity ID of user who submitted the displayed order
`OnBehalfOfCompID` (#115)	Organization ID of user who submitted the displayed order	Organization ID of user who submitted the displayed order
`DeliverToCompID` (#128)	Taker's executing organization ID	Taker's executing organization ID
`Side` (#54)	Buy (1)/Sell (2) from the taker's perspective	Buy (1)/Sell (2) from the taker's perspective
`OrderID` (#37)	Executing taker's order ID	Order ID of user's displayed order
`PartyRole` (#452)=11 (Order Origination Trader)	Originating user ID	None
`PartyRole` (#452)=13 (Order Origination Firm)	Originating organization ID	None

Message Fields

Groups of associated, repeating fields are indicated by shading in the table below.

***Table 4-2** Trade Capture Report Message Fields*

Tag	Field Name	Req'd	Value	FIX Format	Description
35	MsgType	Y	AE	String	MsgType=AE
50	SenderSubID	Y	—	String	Counterparty A legal entity ID of the trade. By convention, this is the taker of the trade. If Counterparty A receives the message, this is Counterparty A's legal entity ID. If Counterparty B or another counterparty receives the message, this is the message receiver's trading party ID or settlement code for Counterparty A's legal entity.
57	TargetSubID	Y	—	String	Counterparty B legal entity ID of the trade. By convention, this is the maker or liquidity provider of the trade. If Counterparty B receives the message, this is Counterparty B's legal entity ID. If Counterparty A or another counterparty receives the message, this is the message receiver's trading party ID or settlement code for Counterparty B's legal entity.
115	OnBehalfOfCompID	Y	—	String	Counterparty B organization ID of the trade. By convention, this is the maker of the trade.
116	OnBehalfOfSubID	N	—	String	When the stream LP trading workflow is used, the organization ID of the stream LP is sent in this field.

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
128	DeliverToCompID	Y	—	String	Counterparty A organization ID of the trade. By convention, this is the taker of the trade.
11	ClOrdID	N	—	String	The unique ID assigned by the client to the order, the `ClOrdID` (#11) of the originating New Order – Single message. This field is only included in the STP message if the organization is configured to include the order ID on the STP download.
12	Commission	N	—	Amt	The previous day's end-of-date (EOD) rate expressed as a mid rate. If USD is one of the currency pairs, the rate should be 1.0 since the USD amount will be used for billing. If USD is not one of the currency pairs, then the rate used should be the one which is used to determine the USD amount e.g. for a EUR/JPY trade, if the dealt currency is EUR, then the EOD rate should be the EUR/USD rate. If the dealt currency is JPY, then the rate should be the JPY/USD rate. If JPY/USD rate is not available, then the rate should be obtained from the USD/JPY rate i.e. JPY/USD rate = 1 divided by the USD/JPY rate.
13	CommType	N	1 (one)=per share	char	
15	Currency	Y	—	Currency	Dealt currency
17	ExecID	Y	—	String	The trading application's trade ID
31	LastPx	Y	—	Price	All in price of the trade. For Spot, the same as `LastSpotRate` (#194). For outrights and swaps, the sum of `LastSpotRate` (#194) and `LastForwardPoints` (#195).
32	LastQty	Y	—	Qty	The dealt amount in terms of `Currency` (#15)
37	OrderID	N	—	String	The order's ID as assigned by the FIX server. This field is only included in the STP message if the organization is configured to include the order ID on the STP download.
38	OrderQty	Y	—	Qty	Dealt amount specified in the `Currency` (#15). Same as `LastQty` (#32).
44	Price	Y	—	Price	All in price of the trade, same as `LastPx` (#31)
55	Symbol	Y	—	String	Currency pair symbol
58	Text	N	—	String	Settlement instructions, if any, that were included with the request to trade. Currently used for RFS workflow only.
60	TransactTime	Y	—	UTCTimestamp	The business execution timestamp (the date and time of the order match) in the format *YYYYMMDD-hh:mm:ss.sss*.
64	FutSettDate	Y	—	LocalMktDate	Specific date of trade in the format *YYYYMMDD*. ■ FX spot, outright: Value date ■ FX swap: Near leg value date
70	AllocID	N	—	String	The ID of the containing portfolio for batch trades. May be empty or not included on the message for non-batch trades.
75	TradeDate	Y	—	LocalMktDate	Trade date in the format *YYYYMMDD*.
119	SettlCurrAmt	Y	—	Amt	Settled amount in terms of `SettlCurrency` (#120)
120	SettlCurrency	Y	—	Currency	Settled currency
150	ExecType	N	5=Replace	char	The state of the included trade

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
167	SecurityType	Y	FOR	String	FOR=Foreign Exchange Contract
192	OrderQty2	N	—	Qty	FX spot, outright: Not applicable FX swap: Far leg dealt amount denominated in `Currency` (#15)
193	FutSettDate2	N	■ Tenor ■ Broken date	LocalMktDate	FX spot, outright: Not applicable FX swap: The far leg value date The field contains either a broken date in *YYYYMMDD* format or a standard tenor symbol: ■ Today ■ TOD ■ ON ■ TN ■ SP ■ SN ■ *n*D: A number of days after the current business date (for example, 1D, 2D, 10D) ■ *n*W: A number of weeks after the current business date (for example, 1W, 2W, 3W) ■ *n*M: A number of months after the current business date (for example, 1M, 2M, 3M) ■ *n*Y: A number of years after the current business date (for example, 1Y, 2Y, 3Y) ■ *n*IMM: The next International Monetary Market (IMM) settlement date. IMM dates are the third Wednesday of the last month of every quarter (March, June, September, December). Entering IMM results in the next IMM date on or after the spot date. Entering 2IMM results in two IMM dates after the spot date. ■ S*n*IMM: (spot + IMM) for swaps ■ T*n*IMM: (tomorrow + IMM) for swaps
194	LastSpotRate	N	—	Spot Rate	■ FX spot: Not applicable. ■ FX outright: Spot rate ■ FX swap: Near-leg spot rate
195	LastForwardPoints	N	—	Forward Points	FX spot: Not applicable FX outright: Forward points FX swap: Near-leg forward points
479	CommCurrency	N	—	Currency	The dealt currency
487	TradeReportTransType	Y	—	char	The message transaction type: N=New, for a new STP deal download R=Replace, for an STP resend, rebook, or a response to a Trade Capture Report Request from the client C=Canceled by Integral (usually as part of portfolio netting and allocation) 4=Reverse (FIX 4.4 only)
541	MaturityDate	See descr.	—	LocalMktDate	The fixing date for NDF trades. Either this field or a value date (`FutSettDate` (#64)) must be specified. If this field is omitted or empty, then the fixing date is calculated by the system. If both this field and a value date are specified, then the value of this field must be before `FutSettDate` (#64).
568	TradeRequestID	N	—	String	Request ID if the message is sent as a response to a Trade Capture Report Request from the client

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
570	PreviouslyReported	See descr.	Y=Resend, rebook, or trade status N=Not previously sent	String	Indicates if the trade capture report was previously reported to the counterparty. Required by FIX Client API if the trade is a resend, rebook, or trade status response. Not required by FIX Client API if the trade has not previously been sent.
571	TradeReportID	Y	—	String	Unique ID for the Trade Capture Report message
578	TradeInputSource	N	—	String	The channel (application, UI component, or API) that originated the trade.
631	MidPx	N	—	Price	The near-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field is empty or 0 (zero). Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
7631	MidPxFl	N	—	Price	The far-leg all-in mid price. If a provider sends only mid swap points instead of near and far mid prices separately, this field contains the swap points. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
640	Price2	N	—	Price	Same as `LastPx2` (#7541)
641	LastForwardPoints2	N	—	Points	FX spot, outright: Not applicable FX swap: Far leg forward points
7541	LastPx2	N	—	Price	FX spot, outright: Not applicable FX swap: Far-leg all-in price. The sum of `LastSpotRate` (#194) and `LastForwardPoints2` (#641).
7545	SettlCurrAmt2	N	—	Amt	FX spot, outright: Not applicable FX swap: Far leg settled amount in `SettlCurrency` (#120)
7549	OrderNotes	N	—	String	Notes on the order entered by the user who submitted the order. Applicable to single-leg trades only. This field is only included on messages that are sent by the user who submitted the order or sent to the user's organization. Maximum 256 characters. Angle brackets "<" and ">" are not allowed in the value of this field.
7571	PPSpotSprd	N	—	String	Price Provisioning Spread: Spot Defined in price provisioning Admin screens. Spot trades: The spread applied to the spot rate as defined in the price provisioning rule. Outright and swap trades: The sum of the tenor based pre-trade and post-trade spreads in price provisioning rule. Applied on the spot rate component. Available to makers who have configured STP to include market rate and spreads.
7572	PPFwdSprd	N	—	String	Price Provisioning Spread: Forward Defined in price provisioning Admin screens. For outright trades only: The spread as defined in the price provisioning rule applied to the forward rate. Available to makers who have configured STP to include market rate and spreads.

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
7573	PPNearSprd	N	—	String	Price Provisioning Spread: Near Leg Defined in price provisioning Admin screens. The fixed spread as defined in the price provisioning rule applied to the spot rate. The value of this attribute is currently set to zero. Available to makers who have configured STP to include market rate and spreads.
7574	PPFarSprd	N	—	String	Price Provisioning Spread: Far Leg Defined in price provisioning Admin screens. The fixed spread as defined in the price provisioning rule applied to the spot rate. Available to makers who have configured STP to include market rate and spreads.
7575	PPCustSprd	N	—	String	Price Provisioning Spread: Customer This is a derived value. The sum of the spot spread (`PPSpotSprd`), the near leg spread (`PPNearSprd`), and the far leg forward spread (`PPFarSprd`). Available to makers who have configured STP to include market rate and spreads.
7576	PMSpotSprd	N	—	String	Price Making Spread: Spot This a derived value. The sum of pre-trade and post-trade fixed spreads applied to the spot rate. This value includes price improvement, if any. Can be negative. Applied based on the side of the trade or the far leg. If bid, the `PMSpotSprd` is subtracted. If offer, the `PMSpotSprd` is added. ■ PMSpotSprd (bid/sell) = Market spot rate – Customer verified spot rate + (PMMinSprd or PMMaxSprd) – PPSpotSprd ■ PMSpotSprd (offer/buy) = Customer verified spot rate – Market spot rate – (PMMinSprd or PMMaxSprd) + PPSpotSprd Available to makers who have configured STP to include market rate and spreads.
7577	PMPreSpotSpread	N	—	String	Price Making Spread: Spot, Pre-Trade Defined in price making Admin screens on the price stream. The pre-trade fixed spread from the price stream configuration applied to the spot rate. This value is applied based on the side of the far leg. If the far leg spot side is bid, the PMPreSpotSpread is subtracted. If the far leg spot side is offer, the PMPreSpotSpread is added. Available to makers who have configured STP to include market rate and spreads.
7578	PMPostSpotSpread	N	—	String	Price Making Spread: Spot, Post-Trade Defined in price making Admin screens on the price stream. The post-trade fixed spread from the price stream configuration applied to the spot rate. This value is applied based on the side of the far leg. If the far leg spot side is bid, the `PMPostSpotSpread` is subtracted. If the far leg spot side is offer, the `PMPostSpotSpread` is added. Available to makers who have configured STP to include market rate and spreads.

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
7579	PMFwdSprd	N	—	String	Price Making Spread: Forward This is a derived value. For outright trades only: The sum of the pre-trade and post-trade fixed spreads applied to the forward points. This value includes price improvement, if any. Available to makers who have configured STP to include market rate and spreads.
7580	PMNearSprd	N	—	String	Near Tenor-Based Fixed Price-Making Spread This is a derived value. The sum of pre-trade and post-trade fixed spreads for the near leg. This value includes price improvement, if any. Available to makers who have configured STP to include market rate and spreads.
7581	PMFarSprd	N	—	String	Far Tenor-Based Fixed Price-Making Spread This is a derived value. The sum of pre-trade and post-trade fixed spreads for the far leg. This value includes price improvement, if any. Available to makers who have configured STP to include market rate and spreads.
7582	PMPreFwdSpread	N	—	String	Price Making Spread: Forward, Pre-Trade Defined in price making Admin screens on the price stream. For outright trades only: The pre-trade fixed spread from the price stream configuration applied to the forward points. Available to makers who have configured STP to include market rate and spreads.
7583	PMPreNearSpread	N	—	String	Price Making Spread: Near, Pre-Trade Defined in price making Admin screens on the price stream. The pre-trade fixed spread from the price stream configuration for the near leg. Available to makers who have configured STP to include market rate and spreads.
7584	PMPreFarSpread	N	—	String	Price Making Spread: Far, Pre-Trade Defined in price making Admin screens on the price stream. The pre-trade fixed spread from the price stream configuration for the far leg. Available to makers who have configured STP to include market rate and spreads.
7585	PMPostFwdSpread	N	—	String	Price Making Spread: Forward, Post-Trade Defined in price making Admin screens on the price stream. For outright trades only: The post-trade fixed spread from the price stream configuration applied to the forward points. Available to makers who have configured STP to include market rate and spreads.
7586	PMPostNearSpread	N	—	String	Price Making Spread: Near, Post-Trade Defined in price making Admin screens on the price stream. The post-trade fixed spread from the price stream configuration for the near leg. Available to makers who have configured STP to include market rate and spreads.
7587	PMPostFarSpread	N	—	String	Price Making Spread: Far, Post-Trade Defined in price making Admin screens on the price stream. The post-trade fixed spread from the price stream configuration for the far leg. Available to makers who have configured STP to include market rate and spreads.

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
7588	PMMaxSprd	N	—	String	Maximum Price-Making Spread This is a derived value. The maximum spread applied to the spot rate. ■ Bid trade (sell) min/max spread = (Market spot rate – total Price Making pre-trade spread – total Price Making post-trade spread – total Price Provisioning spread) – Customer accepted rate ■ Offer trade (buy) min/max spread = Customer accepted rate – (Market spot rate + total Price Making pre-trade spread + total Price Making post-trade spread + total Price Provisioning spread) Available to makers who have configured STP to include market rate and spreads.
7589	PMMinSprd	N	—	String	Minimum Price-Making Spread This is a derived value. The minimum spread applied to the spot rate. ■ Bid trade (sell) min/max spread = (Market spot rate – total Price Making pre-trade spread – total Price Making post-trade spread – total Price Provisioning spread) – Customer accepted rate ■ Offer trade (buy) min/max spread = Customer accepted rate – (Market spot rate + total Price Making pre-trade spread + total Price Making post-trade spread + total Price Provisioning spread) Available to makers who have configured STP to include market rate and spreads.
7590	PPPostSpotSpread	N	—	String	Price Provisioning Spread: Spot, Post-Trade Defined in price provisioning Admin screens. For spot and outright trades: Post-trade fixed spread as defined in the price provisioning rule applied to the spot rate. Available to makers who have configured STP to include market rate and spreads. For swaps, this value is applied based on the side of the far leg.
7591	PPPreSpotSpread	N	—	String	Price Provisioning Spread: Spot, Pre-Trade Defined in price provisioning Admin screens. For spot and outright trades: Pre-trade fixed spread as defined in the price provisioning rule applied to the spot rate. Available to makers who have configured STP to include market rate and spreads. For swaps, this value is applied based on the side of the far leg.
7592	PPPreFwdSpread	N	—	String	Price Provisioning Spread: Forward, Pre-Trade Defined in price provisioning Admin screens. For outright and swap trades only: Pre-trade fixed spread as defined in the price provisioning rule applied to the forward points. Available to makers who have configured STP to include market rate and spreads.

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
7593	PPPostFwdSpread	N	—	String	Price Provisioning Spread: Forward, Post-Trade Defined in price provisioning Admin screens. For outright and swap trades only: Post-trade fixed spread as defined in the price provisioning rule applied to the forward points. Available to makers who have configured STP to include market rate and spreads.
7601	CoverExecID	N	Single trade cover: *tradeID* Multiple trade cover: *tradeID-tradeID-tradeID*	String	The ID of the trade that covers this trade. If more than one trade covered this trade, the IDs are included as a single string with the IDs separated by hyphens (ASCII character 45), for example: FXI41606-FXI41607-FXI41610
7602	CoveredExecID	N	—	String	The ID of the trade that this trade covers.
7901	originatingPortfolioId	N	—	String	For batch trades: The ID of the portfolio that originated this trade. Available for the "FIX4.3 (35=AE)" STP format only. The appearance of the `originatingPortfolioId` (#7901) field in STP messages is controlled by the **Include Originating Portfolio ID** checkbox in the STP Download Configuration screen (**Organization > STP Download Configuration > Edit**) of the Admin Portal. The checkbox is only active when you choose "FIX4.3 (34=AE)" from the **Format** drop-down list.
9374	USINamespace	N	—	String	The ten-character USI namespace for single-leg trades or the near-leg USI prefix for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9375	USIFarNamespace	N	—	String	The ten-character USI namespace for the far-leg. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9376	USI	N	—	String	The USI for single-leg trades or the near-leg USI for multi-legged trades. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9377	USIFar	N	—	String	The far-leg USI. Only included when the receiving organization is configured to receive SEF identifiers and when the trade involves SEF instruments and organizations.
9378	UTIPrefix	N	—	String	UTI Namespace for single-leg trades (outrights) and near leg of swaps
9379	UTIFarPrefix	N	—	String	UTI Namespace for far leg of swaps. Not applicable to single-leg trades.
9380	UTI	N	—	String	UTI value for single-leg trades (outrights) and near leg of swaps
9381	UTIFar	N	—	String	UTI value for far leg of swaps. Not applicable to single-leg trades.
The following shaded rows are a repeating group of fields that represent the sides on the trade. The value of the NoSides (#552) field indicates the number of sides. The required fields must be included as a group for each side.					
552	NoSides	Y	1= One side 2=Both sides	NumInGroup	The number of sides represented in the message

Table 4-2 *Trade Capture Report Message Fields (continued)*

Tag	Field Name	Req'd	Value	FIX Format	Description
54	Side	Y	—	char	The buy/sell side from perspective of the customer specified by the `SenderSubID` (#50) and `DeliverToCompID` (#128) fields in terms of the dealt `Currency` (#15). For swaps, this is the side of the far leg. 1=Buy 2=Sell
37	OrderID	Y	—	String	Unique ID generated by the FIX server
End of repeating group: NoSides (#552)					
The following shaded rows are a repeating group of fields that represent a party on the trade. The value of the NoPartyIDs (#453) field indicates the number of groups and thus the number of parties. The required fields must be included as a group for each party.					
453	NoPartyIDs	Y	3	NumInGroup	The number of repeating groupings of `PartyID` (#448), `PartyIDSource` (#447), and `PartyRole` (#452) to represent the parties on the trade (taker user, originating user, originating organization).
448	PartyID	Y	—	String	The ID of the party specified by the `PartyRole` (#452) of this group of fields.
447	PartyIDSource	Y	D=Proprietary/Custom code	char	The class or source of the `PartyID` (#448) value.
452	PartyRole	Y	3=Client ID 11=Order Origination Trader (associated with Order Origination Firm, trader who initiates/submits the order) 13=Order Origination Firm	Int	The type or role of the associated `PartyID` (#448). 3='Taker user 11=Originating user 13=Originating organization
End of repeating group: NoPartyIDs (#453)					

APPENDIX A

Error Codes

A.1 Error Codes

This appendix lists and describes the various error codes by message type. The error codes are contained by the `Text` (#58) field.

Table A-1 *Error Codes by Message Type*

Message Type	`Error Code`	**Description**
All Message Types	`INTERNAL_SERVER_ERROR`	System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
Logon `MsgType` (#35)=A "Logon" on page 65	`SenderCompIDFormatIncorrect`	`SenderCompID` (#49) format is not correct. It should be `quote.`*`FI-Name`* or `trade.`*`FI-Name`*.
	`UserAuthenticationFailure`	Password is incorrect.
	`UserNameMissing`	User name not set/sent with login message.
	`UserNameSetUp`	User name is wrong or user does not exist.
	`UserNamespaceMismatch`	User name sent in login does not exist for login organization.
	`UserOrganizationSetUp`	User organization sent as part of `SenderCompID` (#49) (`quote.`*`FI-Name`* or `trade.`*`FI-Name`*) does not exist.
	`UserPasswordMissing`	Password is not set/sent with login message.

***Table A-1** Error Codes by Message Type (continued)*

Message Type	Error Code	Description
MarketDataRequest MsgType (#35)=V "Market Data Request" on page 71	CurrencyPair Not Supported	Ccy pair is not supported
	INSUFFICIENT_DATA	Provider related configurations are incomplete.
	INTERNAL_SERVER_ERROR	Request is Null. System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
	INTERNAL_SERVER_ERROR	SendMessage failure to provider. System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
	InverseCurrencyPair.NotSupported	if rate is inverted
	RequestValidationError.channel	Request Channel is Null
	RequestValidationError. CurrencyPairNotSet	Symbol is missing .
	RequestValidationError. CurrencyPairNotSupported	Symbol is not supported.
	RequestValidationError. DeliverToCompIDNotSet	DeliverToCompID (#128) is missing.
	RequestValidationError. DeliverToCompIDSetUp	Liquidity provider organization does not exist. DeliverToCompID (#128) contains ID of the liquidity provider.
	RequestValidationError. DuplicateMDRequestID	The MDReqID (#262) has already been used to subscribe a currency pair or the subscription already exist for this currency pair.
	RequestValidationError. DuplicateSubscriptionRequest	Rate subscription already exists for given currency pair and liquidity provider combination.
	RequestValidationError. IncorrectMDEntryType	MDEntryType (#269) value is not correct. It should be "0" or "1".
	RequestValidationError. InvalidCurrencyPair	Currency pair or currency pair format is invalid.

Table A-1 *Error Codes by Message Type (continued)*

Message Type	Error Code	Description
MarketDataRequest MsgType (#35)=V (continued)	RequestValidationError. MDUpdateTypeNotSupported	MDUpdateType (#265) is not supported. Valid value is "0".
	RequestValidationError. MoreThanOneGroup	Field NoRelatedSym (#146) should be equal to 1.
	RequestValidationError. NoTradingRelationship	The organization does not have a trading relationship with the liquidity provider.
	RequestValidationError.permission : No permission to subscribe to own organization prices	User does not have intrafloor price n trading perm
	RequestValidationError.permission : No permission to trade with own organization prices	User does not have intrafloor trading perm
	RequestValidationError. ProductNotSet	Product is missing.
	RequestValidationError. ProductNotSupported	Product is not supported. Its value should be 4 (Currency).
	RequestValidationError. requestClassfication	Request classification is null
	RequestValidationError. SenderSubIDNotSet	SenderSubID (#50) is missing.
	RequestValidationError. SenderSubIDSetUp	Trading party for given name does not exist in liquidity provider.
	RequestValidationError. SubscriptionRequestTypeNotSupported	SubscriptionRequestType (#263) is not supported. Valid values are "1" for subscription and "2" for unsubscription.
	RequestValidationError.toOrganization	Request does not include toOrg
	RequestValidationError. unEqualNamespace	User's name space doesn't match with that of request organization's namespace.
	TradeValidationError.currencyEqual	Base currency is same as term currency.
MarketDataRequest MsgType (#35)=V (continued)	TradeValidationError.currencyNull	Base currency or term currency is null.
	TradeValidationError.rateBasisNull	FXRateBasis=null

***Table A-1** Error Codes by Message Type (continued)*

Message Type	Error Code	Description
New Order – Single MsgType (#35)=D "New Order – Single" on page 90	INSUFFICIENT_DATA	Provider related configurations are incomplete.
	INTERNAL_SERVER_ERROR	Quote service is null. System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
	INTERNAL_SERVER_ERROR	acceptedQuoteReferrence object is null or request object is null. System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
	INTERNAL_SERVER_ERROR	preAccept/acceptance Failure. System Error, a serious error. Stop trading immediately and contact Business Support to resolve the issue.
	PROVIDER_NOT_AVAILABLE	Provider is inactive.
	RequestValidationError.amount	ccy1amt==0 && ccy2amt==0
	RequestValidationError. BuySellMismatch	Buy/Sell side of the accepted price is incorrect.
	RequestValidationError.channel	Request Channel is Null
	RequestValidationError. DeliverToCompIDNotSet	DeliverToCompID (#128) is missing.
	RequestValidationError. DeliverToCompIDSetUp	Liquidity provider organization does not exist. DeliverToCompID (#128) contains ID of the liquidity provider.
	RequestValidationError. DuplicateOrder	Order already exists for given ClOrdID (#11) and PossDupFlag (#43)/PossResend (#97) is not set on message or set with value equals to "N".
	RequestValidationError. ExecInstNotSupported	The value of ExecInst (#18) needs to be a valid combination of execution instructions. See "Order Execution" on page 30.

***Table A-1** Error Codes by Message Type (continued)*

Message Type	Error Code	Description
New Order – Single `MsgType` (#35)=D (continued)	`RequestValidationError. ExecInstNotSupportedForPegOrders`	For pegged orders (`AlgoParameters` (#7560)=POT) the only valid values of `ExecInst` (#18) are "B" (OK to cross) and "P" (Market peg). See "ST" on page 30.
	`RequestValidationError. HandledInstNotSupported`	Field `HandlInst` (#21) has incorrect value. Valid value is "1".
	`RequestValidationError.InvalidDealtCcy`	Dealt currency is neither base currency not term currency.
	`RequestValidationError.InvalidPrice`	Accepted price is invalid. Zero is invalid price.
	`RequestValidationError.InvalidQuoteID`	The `QuoteID` (#117) of the message is invalid
	`RequestValidationError. LegalEntitySetIncorrectly`	Both `OnBehalfOfSubID` (#116) and `SenderSubID` (#50) are present. Only one of them should be set.
	`RequestValidationError. OnBehalfOfCompIDFormatIncorrect`	`OnBehalfOfCompID` (#115) format is not correct. It should be *`userName@CustomerOrg`*.
	`RequestValidationError. OnBehalfOfCompIDNotSet`	`OnBehalfOfCompID` (#115) is missing.
	`RequestValidationError. OnBehalfOfCompIDSetUp`	Customer organization for given name does not exist.
	`RequestValidationError. OnBehalfOfCompIDUserSetUp`	User for given name does not exist.
	`RequestValidationError. OnBehalfOfSubIDNotSet`	`OnBehalfOfSubID` (#116) is missing.
	`RequestValidationError. OnBehalfOfSubIDSetUp`	Customer organization trading party for given name doesn't exist in liquidity provider.

Table A-1 *Error Codes by Message Type (continued)*

Message Type	Error Code	Description
New Order – Single MsgType (#35)=D (continued)	RequestValidationError. OrderPreviouslySend	Order already exist for given ClOrdID (#11) and PossDupFlag (#43) /PossResend (#97) set on message with value equals to "Y".
	RequestValidationError. OrderTypeNotSupported	OrdType (#40) is not supported. Valid values are "D".
	RequestValidationError.permission: No permission to subscribe to own organization prices	User does not have intrafloor price n trading permission
	RequestValidationError.permission: No permission to trade with own organization prices	User does not have intrafloor trading perm
	RequestValidationError.PriceMismatch	Accepted price is not same as previously quoted price.
	RequestValidationError.QuoteExpired	Quote not found. It has expired. Price is no longer valid.
	RequestValidationError. requestClassfication	Request classification is null
	RequestValidationError. SenderSubIDNotSet	SenderSubID (#50) is missing.
	RequestValidationError. SenderSubIDSetUp	Trading Party for given name doesn't exist in liquidity provider. Valid only in case of direct trading.
	RequestValidationError. SideNotSupported	Side (#54) has incorrect value. Valid values are "1" and "2".
	RequestValidationError. TooLateToEnter	Acceptance has not taken in allowed time period. This error results most frequently from unsynchronized servers. See "Server Synchronization" on page 43.
	RequestValidationError. toOrganization	Request does not include toOrg
	RequestValidationError. tradingDisabled	Org external system id 'READONLY_ORG'='Y' or Trading is disabled

***Table A-1** Error Codes by Message Type (continued)*

Message Type	Error Code	Description
New Order – Single `MsgType` (#35)=D (continued)	`RequestValidationError.unEqualNamespace`	User's name space doesn't match with that of request organization's namespace.
	`TradeValidationError.businessDate`	Value date is not a business date
	`TradeValidationError.currencyEqual`	Base currency is same as term currency.
	`TradeValidationError.currencyNull`	Base currency or term currency is null.
	`TradeValidationError.MaxTenor`	ValueDate is more than the max tenor
	`TradeValidationError.rateBasisNull`	Rate Basis is not set.
	`TradeValidationError.settlementDateRule`	maxTenorRule is null
	`TradeValidationError.valueDate`	Value Date is earlier than Trade Date
Order Status Request `MsgType` (#35)=H "Order Status Request" on page 119	`DataMismatch.BuySell`	Side does not match with requested trade's currency pair.
	`DataMismatch.CurrencyPair`	Symbol does not match with requested trade's currency pair.
	`DataMismatch.DeliverToCompID`	Provider organization ID provided under `DeliverToCompID` (#128) does not match requested trade's provider organization.
	`DataMismatch.LegalEntity`	Trading party name provided under `SenderSubID` (#50) /`OnBehalfOfSubID` (#116) does not match with requested trade's trading party.
	`DataMismatch.OnBehalfOfCompID`	Customer organization name provided under `OnBehalfOfCompID` (#115) does not match requested trade's organization.
	`ErrorRetrievingData`	No order found for given `ClOrdID` (#11). System error.
	`RequestValidationError.LegalEntityNotSet`	Trading party name not set in `SenderSubID` (#50) /`OnBehalfOfSubID` (#116)
Order Status Request `MsgType` (#35)=H (continued)	`UnknownDealState`	Deal state is unknown. Verification is still pending.
	`UnknownOrder`	No order found for given `ClOrdID` (#11).

APPENDIX B

Changes

B.1 Changes

This appendix provides information about the changes to the latest versions of this document. The information is in chronological order from newest to oldest.

Table B-1 *Document Changes*

Date	Enhancements
January 2015 (5.6v1)	■ No changes were made to FIX Client API messages in this release.

***Table B-1** Document Changes (continued)*

Date	Enhancements
April 2015 (5.5v2)	■ STP for batch trades: Added `originatingPortfolioId` (#7901) to "Trade Capture Report" on page 139. ■ STP: Added the following spread values to "Trade Capture Report" on page 139: ☐ `PPSpotSprd` (#7571) ☐ `PPFwdSprd` (#7572) ☐ `PPNearSprd` (#7573) ☐ `PPFarSprd` (#7574) ☐ `PPCustSprd` (#7575) ☐ `PPPostSpotSpread` (#7590) ☐ `PPPreSpotSpread` (#7591) ☐ `PPPreFwdSpread` (#7592) ☐ `PPPostFwdSpread` (#7593) ☐ `PMSpotSprd` (#7576) ☐ `PMPreSpotSpread` (#7577) ☐ `PMPostSpotSpread` (#7578) ☐ `PMFwdSprd` (#7579) ☐ `PMNearSprd` (#7580) ☐ `PMFarSprd` (#7581) ☐ `PMPreFwdSpread` (#7582) ☐ `PMPreNearSpread` (#7583) ☐ `PMPreFarSpread` (#7584) ☐ `PMPostFwdSpread` (#7585) ☐ `PMPostNearSpread` (#7586) ☐ `PMPostFarSpread` (#7587) ☐ `PMMaxSprd` (#7588) ☐ `PMMinSprd` (#7589) ■ Documentation change only, no change to the FIX Client API: Added the following existing fields to "Order Cancel/Replace Request" on page 106 to bring it more closely in line with New Order – Single: ☐ `PossDupFlag` (#43) ☐ `Price`(#44) ☐ `FutSettDate` (#64) ☐ `StopPx` (#99) ☐ `OrigSendingTime` (#122) ☐ `OrderQty2` (#192) ☐ `FutSettDate2` (#193) ☐ `PegOffsetValue` (#211) ☐ `MaturityDate` (#541) ☐ `FarMaturityDate` (#8541) ☐ `ClOrdLinkID` (#583) ☐ `MidPx` (#631) ☐ `OrderNotes` (#7549) ☐ `Price2` (#640) ☐ `ContingencyType` (#1385) ☐ `MidPxFl` (#7631) ☐ `REXParameters` (#7700) ☐ `IsSEFTrade` (#9360) ☐ `ExecutionVenue` (#9361) ☐ `USPerson` (#9362) ☐ `SwapDataRepository` (#9363) ☐ `ClearingVenue` (#9364) ☐ `TakerLEI` (#9365) ☐ `MakerLEI` (#9366) ☐ `IntentToClear` (#9367) ☐ `EndUserException` (#9368) ☐ `IsBlockTrade` (#9369) ☐ `ReportingParty` (#9370) ☐ `USINamespace` (#9374) ☐ `USIFarNamespace` (#9375) ☐ `USI` (#9376) ☐ `USIFar` (#9377) ☐ `UTIPrefix` (#9378) ☐ `UTIFarPrefix` (#9379) ☐ `UTI` (#9380) ☐ `UTIFar` (#9381) ☐ `UPI` (#9382)
January 2015 (5.4v1)	■ No changes were made to FIX Client API messages in this release.

***Table B-1** Document Changes (continued)*

Date	Enhancements
November 2014 (5.3v2)	■ Corrected description of order submission failure workflow. Added order cancel response (Execution Report message with `OrdStatus` (#39)=4 (Canceled)). □ "Order Submission Failure" on page 19 □ Added Figure 1-9, "Order Submission Failure: Cancellation" on page 20. □ Consolidated content from the following sections into the new section "Order Submission Failure" on page 19 and removed the old sections: "Order Submission Failure" from "Execution Report" on page 121, "Application Rejection" and "Business Rejection" from "Post-Order Workflows" on page 19.
October 2014 (5.3v1)	■ SEF: The `MidPxLeg` (#7632) field has been added to the Execution Report message to capture the mid rate for multileg trades. This change applies to the FIX 4.4 format only. See "MidPxLeg" on page 129. Added Execution Report message to example message collection in FIX 4.4 section. ■ SEF: Added support for NDF swaps with `FarMaturityDate` (#8514) added to messages in the RFS workflow: □ Quote Request ("Quote Request" on page 79) □ Quote ("Quote" on page 83) □ New Order – Single ("New Order – Single" on page 90) □ Execution Report ("Execution Report" on page 121) □ Added Quote Request and Execution Report (Fill) messages for NDF swaps to the example messages collection.
September 2014 (5.2.5v1)	■ Added `AllocID` (#70) to the STP Trade Capture Report message for single- and multi-leg trades for reference to trade portfolios for batch trade processing. See "AllocID" on page 141.
August 2014 (5.2v3)	■ SEF: Added `MidPxLeg` (#7632) for multileg trades to "Quote" on page 83. ■ Added `LastSpotRate2` (#7542) to Execution Report message for RFS rejections only. See "LastSpotRate2" on page 130.
August 2014 (5.2v1)	■ For batch trades, added support for undisclosed batch trades (two-way pricing) to "Quote Request" on page 79: □ `Side` (#54): Added value "7=Undisclosed" □ `OrderQty` (#38): Required for undisclosed batch. For disclosed batch, not applicable and must not be included on the request. □ `LegSide` (#624): Added value "7=Undisclosed" □ `LegQty` (#687): For undisclosed batch requests, this value is the net amount for given value date. For example, if two legs for 1W value date with amount 1M buy and 2M sell, then the value of this field for 1W will be 1M. The side of the leg is only revealed after price acceptance. □ `NestedPartyID` (#524): Not required for undisclosed batch. Required for disclosed batch. ■ For batch trades, added support for undisclosed batch trades (two-way pricing) to "Quote" on page 83: □ `LegSide` (#624): Added value "7=Undisclosed" ■ Updated examples with undisclosed batch workflow messages in the FIX 4.4 section of `FIX_Client_API_Examples_5.2v1.docx` packaged with this document.

***Table B-1** Document Changes (continued)*

Date	Enhancements
June 2014 (5.1v1)	■ Added information for order management system (OMS) integration: □ "Integration with an Order Management System (OMS)" on page 42 □ Added `HandlInst` (#21)=3 to New Order – Single message. See "New Order – Single" on page 90. ■ All message examples are now packaged in a Word document instead of separate text files. ■ Removed changes prior to 2012 from this list. ■ Added `OrderNotes` (#7549) to "New Order – Single" ("OrderNotes" on page 97) and "Trade Capture Report" ("OrderNotes" on page 143). ■ SEF: Added the following fields to messages to support pricing SEF requests over the FIX Client API: □ "Execution Report" on page 121: `MakerLEI` (#9366) □ "New Order – Single" on page 90: `IsSEFTrade` (#9360), `ExecutionVenue` (#9361), `USPerson` (#9362), `SwapDataRepository` (#9363), `ClearingVenue` (#9364), `TakerLEI` (#9365), `MakerLEI` (#9366), `IntentToClear` (#9367), `EndUserException` (#9368), `IsBlockTrade` (#9369), `ReportingParty` (#9370), `UPI` (#9382)
March 2014 (5.0v1)	■ Changed required status of `MaturityDate` (#541) on Market Data Snapshot/Full Refresh message from "Y" to "N". See "Market Data Snapshot/Full Refresh" on page 75.

***Table B-1** Document Changes (continued)*

Date	Enhancements
February 2014 (4.13v2)	■ Removed all references to the Market Data – Incremental Refresh message. This message has been deprecated. ■ Clarified optional status of Execution Report messages with New (`ExecType` (#150)=0) state: □ "Order Workflow" on page 13 □ Table 1-2 on page 15 □ "Order Status" on page 36 □ "ExecType" on page 127 ■ All references to "strategy" orders have been changed to "algo" orders. ■ Removed obsolete section "Transaction Sides" from the Trade Capture Report message section. The `NoSides` (#552) field is required and is not omitted. See "Trade Capture Report" on page 139. ■ EMIR: The following UTI fields have been added to these messages Quote Request, Trade Capture Report, New Order – Single, Execution Report: □ `UTIPrefix` (#9378) □ `UTIFarPrefix` (#9379) □ `UTI` (#9380) □ `UTIFar` (#9381) ■ For batch trades, the following corrections and clarifications have been made: □ The value of `ClOrdID` (#11) on the New Order – Multileg message must be the same as the `QuoteReqID` (#131) on the originating Quote Request and Quote messages. □ Each trade in the batch has its own spot rate and forward points information in the Quote message: `LegBidPx` (#681), `LegOfferPx` (#684), `LegBidForwardPoints` (#1067), `LegOfferForwardPoints` (#1067). □ Non-batch pricing fields are not included in the batch responses like the Quote and Execution Report messages, for example `Side` (#54), `BidPx` (#132), `OfferPx` (#133), `SpotRate` (#190), `BidSize` (#134), `OfferSize` (#135), `BidSpotRate` (#188), `BidForwardPoints` (#189), `OfferSpotRate` (#190), `OfferForwardPoints` (#191). □ Non-batch pricing fields on batch Quote Request messages are ignored. □ The order of fields in the `NoLegs` (#555) repeating group differs in the Quote Request and New Order – Multileg messages. See "NoLegs" on page 81 and "NoLegs" on page 104. □ The order of fields has been corrected in the New Order – Multileg message. See "New Order – Multileg" on page 103. □ The tag number of the `NestedPartyID` (#524) field has been corrected in the Quote Request message. □ Added `Product` (#460) to New Order – Multileg message. □ Removed `LegQty` (#687) from Quote, New Order – Multileg, and Execution Report messages. □ Removed "Empty=2-way" value from `LegSide` (#624) on Quote Request and Quote messages. Two-way pricing is currently not supported for batches. □ Removed the following fields from the New Order – Multileg message: `QuoteID` (#117), `Currency` (#15), `SecurityType` (#167). □ Replaced repeating group `NoNestedPartyIDs` (#539) with `LegExecID` (#1893) on Execution Report message. □ EMIR: UTI fields `UTIPrefix` (#9378) and `UTI` (#9380) added to the `NoLegs` (#555) repeating group in batch Quote Request, New Order – Multileg, and Execution Report messages.

***Table B-1** Document Changes (continued)*

Date	Enhancements
November 2013 (4.12v1)	■ FIX: Added the following values to `TradeReportTransType` (#487) on Trade Capture Report message. See "TradeReportTransType" on page 142 and "Message Synchronization" on page 139. □ "C" □ "4" (FIX 4.4 only)
September 2013 (4.11v1)	■ Corrected references to position management messages in Table 1-1, "Supported Message Types" on page 51.
July 2013 (4.10v2)	■ Heartbeat interval of 30 seconds or longer recommended for `HeartBtInt` (#108) on the Logon message. See "HeartBtInt" on page 66. ■ Clarification of message workflow for previously quoted orders. Previously quoted orders do not receive cancellation messages for partial fills. See "Immediate or Cancel (IOC)" on page 32. ■ Updated table of supported messages in "Supported Message Types" on page 50.

***Table B-1** Document Changes (continued)*

Date	Enhancements
May 2013 (4.9v1)	■ Enhanced description of `ExecInst` (#18)=R for stops. See Table 1-3, "Execution Instructions and Applicable Order Types" on page 30. ■ Enhanced description of stop price field (`StopPx` (#99)) on New Order – Single message. See "StopPx" on page 94. ■ Corrected required status of `TradeRequestID` (#568) to "Y" (required) in "Trade Capture Report Request" on page 138. ■ Added overview for mid-mark and USI. See "USIs and Mid Mark Data" on page 48. ■ Added mid-mark information (`MidPx` (#631) and `MidPxFl` (#7631)) to the following messages: □ "Quote" on page 83 □ "New Order – Single" on page 90 □ "Execution Report" on page 121 □ "Trade Capture Report" on page 139 ■ Added USI information (`USINamespace` (#9374), `USIFarNamespace` (#9375), `USI` (#9376), `USIFar` (#9377))to the following messages: □ "Quote Request" on page 79 □ "New Order – Single" on page 90 □ "Execution Report" on page 121 □ "Trade Capture Report" on page 139 ■ Batch trade support: □ Overview of new workflow added. See "Batch Trade Workflow" on page 18. □ New message type. See "New Order – Multileg" on page 103. □ Fields added to "Quote Request" on page 79: `Account` (#1), `NoLegs` (#555), `LegSymbol` (#600), `LegSide` (#624), `LegCurrency` (#556), `LegQty` (#687), `LegSettDate` (#588), `NoNestedPartyIDs` (#539), `NestedPartyID` (#448), `LegRefID` (#654) □ Description of `QuoteReqID` (#131) changed on "Quote Request" on page 79 . □ Description of `NoRelatedSym` (#146) changed on "Quote Request Reject" on page 82 . □ Fields added to "Quote" on page 83: `LegSymbol` (#600), `LegQty` (#687), `LegSide` (#624), `LegCurrency` (#556), `LegSettDate` (#588), `LegBidPx` (#681), `LegRefID` (#654), `LegOfferPx` (#684), `LegBidFwdPoints` (#1067), `LegOfferFwdPoints` (#1068). □ Description of `QuoteReqID` (#131) and `QuoteType` (#537) changed on "Quote" on page 83 . □ Description of `QuoteID` (#117) and `QuoteCancelType` (#298) changed on "Quote Cancel" on page 87. □ Fields added to "Execution Report" on page 121: `ExecTransType` (#20), `NoLegs` (#555), `LegSymbol` (#600), `LegSide` (#624), `LegCurrency` (#556), `LegQty` (#687), `LegPrice` (#566), `LegSettDate` (#588), `NoNestedPartyIDs` (#539), `NestedPartyID` (#524), `LegRefID` (#654), `LegPrice` (#566). □ Description of `Product` (#460) changed on "Execution Report" on page 121. □ Description of the Business Message Reject message updated to include the batch trade workflow. See "Business Message Reject" on page 133.

***Table B-1** Document Changes (continued)*

Date	Enhancements
March 2013 (4.8.2v1)	■ For the ESP workflow, the "Pending New" Execution Report message (`ExecType` (#150)=A) is no longer sent by the server as a response to a New Order – Single submitted by the client. The related workflow diagrams and message field descriptions have been updated to remove references to the "Pending New" message: □ Figure 1-2 on page 14 □ Figure 1-3 on page 14 □ Figure 1-4 on page 15 □ 1.2.3 on page 12 □ Figure 1-10 on page 20 □ Figure 1-11 on page 21 □ Figure 1-12 on page 21 □ Figure 1-13 on page 22
March 2013 (4.8v1)	■ No changes were made to FIX Client API messages in this release. ■ Added the FSR trade type. See "Supported Trade Types" on page 25.
January 2013 (4.7v1)	■ No changes were made to FIX Client API messages in this release. ■ Updated requirement that `DeliverToCompID` (#128) must be set to "ALL" in the request message to receive incremental updates. The behavior for any value other than "ALL" when requesting incremental updates is not defined. See "DeliverToCompID" on page 73 and "Market Data – Incremental Refresh (FX Inside to Client)" on page 118 ■ Removed mention of even number requirement for `NoMDEntries` (#268) values in all messages. ■ Moved `NoMDEntries` (#268) description so that it is clear that the field is not part of the repeating group. See "NoMDEntries" on page 76 and "NoMDEntries" on page 119. ■ Updated description of `MDEntryPx` (#270) to clarify client behavior when the field's value is 0 (zero) in Market Data – Incremental Refresh messages. See "MDEntryPx" on page 120.
October 2012 (4.6v1)	■ Updated server synchronization recommendation to use the NTP pool in "Server Synchronization" on page 43. ■ Removed incorrect requirement for new refresh message for order placement in "Multi-tier Quotes" on page 45.
September 2012 (4.5.1v2)	■ The section describing order cancel/replace and order amendment has been rewritten. See "Order Cancel/Replace and Amendment" on page 28. ■ The description of IOC and FOK times in force have been expanded to clarify fill size requirements and the effect of the `MinQty` (#110) field. See "Immediate or Cancel (IOC)" on page 32 and "Fill or Kill (FOK)" on page 32. ■ The behavior of the Order Cancel/Replace workflow has changed for partially filled orders. This change does not apply to the Order Amendment workflow. If an Order Cancel/Replace Request message is received by the system for an order that is partially filled, the request is rejected with an Order Cancel Reject Message with the `CxlRejReason` (#102) field set to 0 (zero) "Too Late To Cancel". See "Order Cancel/Replace and Amendment" on page 28.

***Table B-1** Document Changes (continued)*

Date	Enhancements
August 2012 (4.5v2)	■ The `TransactTime` (#60) field on Execution Report messages for fills and Trade Capture Report messages for STP now contains the business execution date (the date and time of the order match). See "TransactTime" on page 125 and "TransactTime" on page 141. ■ The value of the `MinQty` (#110) field can now affect whether or not an order is a displayed order: □ "Order Visibility" on page 33 □ "MinQty" on page 95 □ "MinQty" on page 109 ■ Corrected description of `MinQty` (#110). For orders allowing partial fills, subsequent fills can be of any size. □ "Partial Fills" on page 32 □ "MinQty" on page 95 □ "MinQty" on page 109 ■ Updated description `ExpireTime` (#126) for GTD expiry type to be a relative time instead of GMT. See "Good Till Date/Time (GTD)" on page 32. ■ Changed `Currency` (#15) on Quote Request to be a required field. See "Currency" on page 80. ■ The `MaxShow` (#210) field on a New Order – Single message must either be included in the message with a value or left out of the message completely. If the field is included with no value, the message will be rejected. See "Order Visibility" on page 33 and "MaxShow" on page 96. ■ Removed obsolete `RequestValidationError.LegalEntityNotSet` error message from "Error Codes" on page 149.
July 2012 (4.4.5v2)	■ Added `MinQty` (#110) to repeating groups of rates on the following messages: □ "Market Data Snapshot/Full Refresh" on page 75 □ "Market Data – Incremental Refresh (FX Inside to Client)" on page 118 ■ When the stream LP trading workflow is used, the anonymous ID of the stream LP is sent in the `OnBehalfOfSubID` (#116) field of the Execution Report message. See "OnBehalfOfSubID" on page 122. ■ Customers can now request aggregated rates for term currency trading. See "Currency" on page 75.
June 2012 (4.4v1)	■ Changed all references to "undisclosed LP" to "stream LP". ■ As an optimization for the order cancel/replace workflow and order amendment, the server now sends two Execution Report messages instead of four messages as in previous releases. See "Order Amend and Cancel/Replace Workflow" on page 16. ■ Updated description of algo order parameters: "PegTime" on page 39, "TOBPercent" on page 40, "PegOffsetType" on page 41, ■ Added Amend and Cancel/Replace workflow states to order state transition overview. See Figure 1-1 on page 38. ■ Added information about liquidity provider names with merged prices in aggregated books. See "Provider Names with Merged Prices and Provider Priority" on page 45.

***Table B-1** Document Changes (continued)*

Date	Enhancements
May 2012 (4.3v4)	■ Removed incorrect reference to `ExecID` (#17) in "Message Synchronization" on page 139. ■ When the undisclosed LP trading workflow is used, the ID of the undisclosed LP is sent in the `OnBehalfOfSubID` (#116) field of the Trade Capture Report message: □ Table 1-9, "Org and User IDs for Direct Users: Trade Capture" on page 60 □ "Trade Capture Report" on page 139 ■ STP download has been enhanced to allow for better routing and representation of counterparties in intrafloor trades. See "Intrafloor Trades" on page 139. ■ For order cancellation, the value of the `LeavesQty` (#151) field in the Execution Report message response with `ExecType` (#150)=6 (PendingCancel) is now deterministic and no longer needs to be disregarded. See "LeavesQty" on page 127. ■ Added the `MinQty` (#110) field to the Market Data Request message. A quote with a size smaller than the `MinQty` (#110) value is not included in an aggregated price. See "MinQty" on page 75.
May 2012 (4.3v4)	■ The system now supports OUO orders: □ "One-Updates-the-Other (OUO) Orders" on page 35 □ Added values to ContingencyType (#1385) in New Order – Single and Execution Report. See "ContingencyType" on page 98 and "ContingencyType" on page 130. ■ Corrected state transition diagram to include New > Filled transition. See Figure 1-1, "Order State Transitions " on page 38. ■ Clarified behavior of `ExecInst` (#18) on Order Cancel/Replace Request messages: values must be resubmitted to persist the values of the original order. See "ExecInst" on page 107. ■ Corrected list of values for `ExecType` (#150) on Execution Report to include 1=Partial Fill and 2=Fill. See "ExecType" on page 127. ■ Clarified required status of `MaxShow` (#210). The field is not required. If the field is not included on a message, then the order is a hidden order. See "Order Visibility" on page 33, "New Order – Single" on page 90, "Order Cancel/Replace Request" on page 106, "Execution Report" on page 121.
April 2012 (4.2v2)	■ Significant reworking of "Order Cancel/Replace and Amendment" on page 28, clarifying order states and the effects of amending by amount and rate. ■ Added support for NDF trades: □ Added trade type to "Supported Trade Types" on page 25. □ Added NDF to `OrdType` (#40)=D in "Quote Request" on page 79 and "New Order – Single" on page 90. □ Added `MaturityDate` (#541) to "Quote Request" on page 79, "Quote" on page 83, "New Order – Single" on page 90, "Execution Report" on page 121, and "Trade Capture Report" on page 139. ■ Added "Stopped" order state: □ Added Figure 1-4. "Order Trading Workflow (Algo Order with Delayed Start) " on page 15. □ Modified Execution Report in Table 1-2, "Order Trading Messages" on page 15 to include algos with delayed start and suspended orders. □ Added "Stopped" state to Figure 1-1. "Order State Transitions " on page 38 and Table 1-4, "Order Status" on page 38. □ Added "Stopped" state to `OrdStatus` (#39) in "Execution Report" on page 121.

***Table B-1** Document Changes (continued)*

Date	Enhancements
February 2012 (4.1.5v8)	■ The `MessageType` (#35) values for the following messages have been corrected: □ "Request for Positions" on page 134 □ "Request for Positions Ack" on page 135 □ "Positions Report" on page 136 ■ Added "Order Cancel/Replace and Amendment" on page 28. ■ Added explanation of Order Cancel/Replace workflow. See "Order Amend and Cancel/Replace Workflow" on page 16. ■ The section "Multiple Execution Attempts" has been changed to "Duplicate Order IDs" on page 34 with a focus on avoiding duplicate `ClOrderID` (#11) values for certain order states. ■ Clarified requirement status of the `Price` (#44) field on New Order – Single message (not required for market orders). See "Price" on page 93. ■ Updated description of `NoDates` (#580), `TradeDate` (#75), `TransactTime` (#60) to reflect query by trade date/time functionality. See "Trade Capture Report Request" on page 138. ■ Removed `Channel` (#7603) and replaced with `TradeInputSource` (#578) in "Trade Capture Report" on page 139. ■ Added `ClOrdID` (#11) and `OrderID` (#37) to "Trade Capture Report" on page 139. ■ Corrected number of `AggregationType` field from #7547 to #7548 in "Market Data Request" on page 71. ■ Clarified the description of "PegOffsetValue" on page 97. ■ The `SettlCurrAmt` (#119) and `SettlCurrency` (#120) fields are optional in the Execution Report message to account for scenarios that do not involve a settled currency, such as cancelled orders. See "SettlCurrAmt" on page 126 and "SettlCurrAmt" on page 126. ■ Renamed section "Order Rejection Reasons" to "Order Submission Failure" on page 19 and changed references to "order rejection" to "order submission failure". ■ Clarified the value of `LeavesQty` (#151) with regards to the order state as indicated by `ExecType` (#150). See "Execution Report" on page 121. ■ Removed `ExecutionReport_TradeRejection.txt` file from example suite (obsolete). ■ The New Order – Single, Order Cancel/Replace Request and Execution Report messages have been enhanced to support algo orders: □ Added overview of algo orders. See "Algo Orders" on page 38. □ `ExecInst` (#18)=ST (see "ST" on page 30) □ `EffectiveTime` (#168) now specifies the absolute time in GMT at which the algo should start execution when `ExecInst` (#18)=ST. □ The following fields have been added: `ExecEndTime` (#7556), `AlgoParameters` (#7560), `AlgoName` (#7561), `ExecEffPeriod` (#7564), `ExecEndPeriod` (#7565). See "New Order – Single" on page 90, "Order Cancel/Replace Request" on page 106, and "Execution Report" on page 121. □ Added `RequestValidationError.ExecInstNotSupportedForPegOrders` to "Error Codes" on page 149. ■ Alphabetized error codes for easier reference. See "Error Codes" on page 149.
August 2006	First version