FXInside™

# ESP FIX Provider API Guide

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



Monday, April 06, 2015 2.1v2

# Contents

# Foreword

## About the Integral FX Grid®

The Integral FX Grid is 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. The Integral 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.

## About This Document

This document describes the Integral FIX Provider API that liquidity providers use to access the Integral FX Grid.

### Organization of This Document

The following chapters discuss the configuration of the FIX Provider API, and the trading workflows and messages that the FIX Provider API supports:

- "FIX Solution" on page 9: The FIX implementation, business rules, and overviews of messages and workflows
- "FIX Session Management" on page 17: The messages used to control the FIX session and manage message conversations
- "Trading Workflows" on page 22: Business workflows and message conversations described
- "Trading Messages" on page 33: Trading message structure and data

In addition, the appendix "Changes" on page 46 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.

***Table 4-1** Character Formatting Conventions*

| Format | Description |
|---|---|
| *Italic* | New terms (the *system monitoring* service) |
| **Bold** | User-interface elements (the **Update** button) |
| `Sans Serif` | ■ Names of classes, instances, messages, and examples of code (the `Counterparty` class)<br>■ Filenames, pathnames, commands, and other operating-system constructs (the `/cust/usr` directory) |
| *`Italic Sans Serif`* | Variable elements for which you must substitute a value (the *`yourFilename`*`.xml` file) |
| Blue color | URLs and cross-references that you can click when viewing the document online ("Foreword" on page 7) |

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

***Table 4-2** Structural Conventions*

| Format | Description |
|---|---|
| `OrdStatus` (#39) | The FIX field name with the FIX field number in parentheses |
| *`PathnameRoot`* | Root path references<br>In file names, a directory name that ends with "Root" is a variable representing a root path that depends on your installation.<br>For example:<br>*`IntegralProductRoot`*`/broker/appLogs/`<br>could refer to the path:<br>`/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<br>For example:<br>`anObjectWithVeryLongName.`<br>`methodOnObject`<br>To avoid confusion between different protocols and programming languages, this document does not use special symbols to identify the break unless the protocol or language provides such a mechanism. |

CHAPTER 1

# FIX Solution

## 1.1 Introduction

This document describes the Integral FIX Provider API, a FIX-based channel that allows liquidity providers to offer foreign exchange liquidity to customers and execute trades through the FX Grid.



## 1.2 FIX Implementation

The interface defined by the FX Grid FIX Provider API conforms to the FIX 4.3 specifications. The FIX Protocol Organization provides a complete reference to the protocol at: http://www.fixprotocol.org

### 1.2.1 Supported Message Types

The FX Grid FIX Provider API supports the following FIX messages types:

***Table 1-1*** *Supported Message Types*

| Message Type | Message Name | Inbound to Provider (I), Outbound from Provider (O), or Bidirectional (B) |
|---|---|---|
| **Session Management Messages** | | |
| A | Logon on page 17 | B |
| 5 | Logout on page 18 | B |
| 0 (zero) | Heartbeat on page 18 | B |
| 1 | Test Request on page 19 | B |
| 2 | Resend Request on page 19 | B |
| 3 | Session-Level Reject on page 20 | B |
| 4 | Sequence Reset on page 20 | B |
| **Trading Messages** | | |

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

| Message Type | Message Name | Inbound to Provider (I), Outbound from Provider (O), or Bidirectional (B) |
|---|---|---|
| g | Trading Session Status Request on page 34 | I |
| h | Trading Session Status on page 34 | O |
| V | Market Data Request on page 35 | I |
| W | Market Data Snapshot/Full Refresh on page 36 | O |
| Y | Market Data Request Reject on page 38 | O |
| R | Quote Request on page 33 | I |
| AG | Quote Request Reject on page 33 | O |
| S | Quote on page 33 | O |
| Z | Quote Cancel on page 33 | O |
| D | New Order – Single (previously quoted) on page 38 | I |
| OT | Order Timeout on page 41 | I |
| Q | Don't Know Trade on page 41 | I |
| 8 | Execution Report on page 42 | O |
| j | Business Message Reject on page 45 | B |

## 1.2.2 Ignored and Unsupported Fields

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

Some fields are required conditionally based on the state of the message workflow or on the value of other fields in the message. These conditions and dependencies are clearly indicated in the field descriptions.

## 1.2.3 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 Provider API does not impose a maximum length on undefined `String` fields, such as free-form text fields and ID fields.

### 1.2.4 Message Length

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

### 1.2.5 Encryption

The FIX Provider API does not support FIX message encryption. Instead, the network transport mechanism (VPN, Radianz, or SSL) ensures message security.

### 1.2.6 Message Headers and Footers

FIX engines set a message's header and footer fields automatically according to the message type and the application context as defined by the application's configuration (see "Configuration" on page 12). For these reasons, the standard header and footer as defined by the FIX protocol are not discussed in detail in this document beyond the expected value of the `MsgType` (#35) and the fields that identify message and business senders and targets, such as the `SenderCompID` (#49), and `TargetCompID` (#56) fields.

***Table 1-2** 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 9 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<br>N=Original transmission | Boolean | Indicates possible retransmission of message with this sequence number. |
| 49 | SenderCompID | Y | — | String | The message sender's ID. See "Organization Identification" on page 12. |
| 56 | TargetCompID | Y | — | String | The message target's ID. See "Organization Identification" on page 12. |
| 52 | SendingTime | Y | — | UTCTimestamp | The time of message transmission (GMT). |
| 122 | OrigSendingTime | N | — | UTCTimestamp | Original time of message transmission when transmitting orders as the result of a resend request. |

***Table 1-3** Standard Message Footer 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. |

# 1.3 Configuration

The following sections describe the details involved in establishing a connection via the FIX Provider API.

## 1.3.1 Deployment

Integral deploys a FIX Provider API FIX engine instance for each liquidity provider. The instance is called the FIX Provider API adaptor. The adaptor handles all streams from the liquidity provider and submits orders to the liquidity provider.

## 1.3.2 Network Connectivity

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

- Radianz
- IPSec VPN
- SSL

## 1.3.3 Organization Identification

The values of the `TargetCompId` (#49) and `SenderCompId` (#56) fields in the message header are supplied by Integral for each provider connection. The FX Grid does not allow multiple connections using the same `TargetCompId` (#49) and `SenderCompId` (#56) fields.

### Account IDs

Integral is a trading facilitator. Trades are booked directly between the your organization (liquidity provider) and the end customer. You must assign an account ID to each of the provider's customer legal entities trading through the FX Grid. The account ID is set on the `Account` (#1) field of New Order – Single message. The account ID value is agreed upon by your organization and Integral.

The ClientTag on a trade request, if any, is captured in the `PartyID` (#448) field. The ClientTag is a unique ID whose value is set by Integral. It is an optional field. See "New Order – Single (FX Grid to Provider)" on page 38.

### Liquidity Provider Organization ID

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

*sessionType*.*orgShortName*

price.yourBank4

***Table 1-4*** *Organization ID Format*

| Token | Example | Description |
|---|---|---|
| *sessionType* | **price**.yourBank4 | For a description of the following session types, see "Sessions" on page 14:<br>■ price<br>■ order |
| *orgShortName* | price.**yourBank4** | Your organization's short name ID |

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

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

### FX Grid Server ID

Like your organization ID, the FX Grid server ID looks like an Internet domain name:

*environment*.fxgrid

demo.fxgrid

***Table 1-5*** *Organization ID Format*

| Token | Example | Description |
|---|---|---|
| *environment* | **demo**.fxgrid | The specific environment to which you are connected (for example, staging or production) |
| provider | demo.**fxgrid** | The FX Grid server ID |

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

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

## 1.3.4 Connection Mode

The FIX Provider API supports initiator and acceptor connection modes. Communication starts with a Logon message ("Logon (Bidirectional)" on page 17) and ends with a Logout message ("Logout (Bidirectional)" on page 18).

### Acceptor Mode

In the acceptor mode, your FIX engine initiates the FIX connection. The acceptor mode is the default, preferred connection mode for the FIX Provider API. The following parameters must be specified in the FIX configuration of the your FIX engine:

***Example 1-1*** *Parameters for Acceptor Connection Mode*

```
ConnectionType=acceptor
SocketAcceptPort=<Integral's Port>
```

### Initiator Mode

In the initiator mode, the FX Grid initiates the FIX connection to the provider's FIX engine. To configure your systems for acceptor mode, please contact Integral Business Support. The following parameters must be specified in the FIX configuration of your FIX engine:

***Example 1-2*** *Parameters for Initiator Connection Mode*

```
ConnectionType=initiator
SocketConnectHost=<Liquidity Provider's IP>
SocketConnectPort=<Liquidity Provider's Port>
HeartBtInt=5
```

## 1.4 Business Rules

The following business rules apply to all workflows of the FX Grid FIX Provider API.

### 1.4.1 Sessions

The FIX Provider API distinguishes between two session types to optimize your trading message flows. You must establish a both sessions to initiate all trading workflows.:

- Price: Session for sending/receiving market data and quotes. Messages are time-sensitive and transient to enable the high message volume typically associated with prices and price streams. The FX Grid does not resend price session messages in response to a resend request from the provider system. You must establish a price session to initiate all trading workflows. See "Starting the Trading Session" on page 22.
- Order: Session for order submission and trade execution. Messages are transactional and persistent with no lost messages allowed, reflecting their business criticality. The FX Grid resends order session messages in response to a resend request from the provider system. You must establish an order session to initiate all trading workflows. See "Starting the Trading Session" on page 22. How you use an order session depends on the trading workflow you employ.

You must log on and establish each FIX session separately with the FX Grid. 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 "Liquidity Provider Organization ID" on page 13.

### 1.4.2 Business Day End and Start

The business day start and end time must be agreed upon by your organization and Integral. A typical daily configuration starts the day at 17:00:05 EST and ends at 17:00:00 EST.

***Example 1-3*** *Daily start/stop times*

```
ResetOnLogout=N
ResetOnLogin=N
ResetOnDisconnect=N
StartTime=17:00:05
EndTime=17:00:00
TimeZone=America/New_York
```

The FIX Provider API allows you to set custom start and end times for every day of the week.

> **NOTE:** The following examples illustrate the capabilities of FIX Provider API start/stop times. Please contact Integral Business Support to configure customer start/stop times.

***Example 1-4*** *Custom start/stop times format*

```
CustomScheduleTime=[startDay0 startTime0]-[endDay endTime]~[startDay1
startTime1]-[endDay1 endTime1]~…
```

***Example 1-5*** *Customer start/stop times example*

```
CustomScheduleTime=[Sunday 08:30:00]-[Monday 06:35:00]~[Monday 06:40:00]-
[Tuesday 04:00:00]~…
```

### 1.4.3 Sequence Number Reset

Your FIX engine must be configured to reset the sequence number `MsgSeqNum` (#34) only at the end of day and not on disconnect.

To reset sequence numbers once a day after the business day end, the following session-level FIX parameters need to be set on the acceptor and initiator side with the following values (assuming 17:00:05 EST and 17:00:00 EST are the start and end time). The initiator side has a slightly larger down-time window to ensure the when the initiator starts the connection, the acceptor is ready to handle the connection. See "Connection Mode" on page 13 for more information about initiator and acceptor connection modes.

***Example 1-6*** *Parameters for Acceptor Sequence Number Reset*

```
ResetOnDisconnect=N
ResetOnLogout=N
StartTime=17:00:05
EndTime=17:00:00
```

***Example 1-7*** *Parameters for Initiator Sequence Number Reset*

```
ResetOnDisconnect=N
ResetOnLogout=N
StartTime=17:00:05
EndTime=17:00:00
```

### 1.4.4 Event Sequencing

The FIX Provider API handles trading events from customers and liquidity providers on a first-come first-served basis.

### 1.4.5 Supported Deal Types

The FIX Provider API currently supports dealing in the following deal types:

- FX spot only

### 1.4.6 Mid Mark Price

The FIX Provider 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. Only required when the receiving organization is required to receive Pre-Trade Mid-Market Mark. This decision is made by the liquidity provider. |
| 7631 | MidPxFl | N | — | Price | The far-leg all-in mid price. Only required when the receiving organization is required to receive Pre-Trade Mid-Market Mark. This decision is made by the liquidity provider. |

CHAPTER 1

# FIX Session Management

## 1.5 Session Management Messages

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



## 1.6 Logon (Bidirectional)

The Logon message is sent by the provider system to start a session with the FX Grid and sent by the FX Grid in response to a valid log-on request.

> **NOTE:** You should reset the FIX sequence number for the price session (`ResetSeqNumFlag` (#141)=Y) after 3 or 4 unsuccessful log-on attempts. Do not reset the sequence number of the order session.

If the FX Grid receives a Logon message with invalid fields, it sends a Logout message in response. See "Logout (Bidirectional)" on page 18.

***Table 1-1** Logon Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | A | String | A=Logon |
| 98 | EncryptMethod | N | 0 | int | 0 (zero)=no encryption is used. This field may be omitted.<br>Radianz or VPN is used for transport-level security. |
| 108 | HeartBtInt | Y | 5 | int | Heartbeat interval in seconds. If `HeartBtInt` is set zero, then no heart beat message is required. The recommended heartbeat interval is 5 seconds. |

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

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 141 | ResetSeqNumFlag | N | Y | Boolean | Indicates that the both sides of the FIX session should reset sequence numbers. You should reset the FIX sequence number for the price session after 3 or 4 unsuccessful log-on attempts. Do not reset the sequence number of the order session.<br>Y=Yes, reset sequence numbers<br>N=No, do not reset sequence numbers |

For examples of this message, see "Examples" on page 50.

# 1.7 Logout (Bidirectional)

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

If the 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 1-2** Logout Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 5 | String | 5=Logout |
| 58 | Text | N | — | String | The reason for the Logon rejection. Only included for incoming Logout messages (FX Grid to provider) in response to a invalid Logon message. Not valid for outgoing Logout messages (provider to FX Grid). |

Please refer to the `Integral_FIX_Provider_API_Examples.docx` file for examples of this message for both order and price sessions.

# 1.8 Heartbeat (Bidirectional)

Both the provider system and the FX Grid send the Heartbeat message to indicate that the connection is active.

The provider system generates a regular heartbeat at the interval defined by the `HeartBtInt` (#108) field in the Logon message from the FX Grid or as a response to a Test Request message from the FX Grid. The recommended heartbeat interval is 5 seconds.

***Table 1-3** Heartbeat Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 0 | String | 0 (zero)=Heartbeat |

***Table 1-3** Heartbeat Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 112 | TestReqID | N | — | String | Required when the heartbeat is the result of a Test Request message. See "Test Request (Bidirectional)" on page 19. |

For examples of this message, see "Examples" on page 50.

# 1.9 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 (Bidirectional)" on page 18.

***Table 1-4** Test Request Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 1 | String | 1=Test Request |
| 112 | TestReqID | Y | — | String | The resulting Heartbeat message contains this ID. The `TestReqId` should be incremental. |

For examples of this message, see "Examples" on page 50.

# 1.10 Resend Request (Bidirectional)

The FX Grid accepts Resend Request messages only for order sessions. Resend Requests sent on a price session are ignored. Both the provider system and the FX Grid can send a Resend Request message when they detect a message gap to request either a single message or a range of messages.

***Table 1-5** Resend Request Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 2 | int | 2=Resend Request |
| 122 | OrigSendingTime | N | — | UTCTimestamp | Original time of message transmission when transmitting orders as the result of a resend request |
| 7 | BeginSeqNo | Y | — | SeqNum | First message of range (inclusive) |
| 16 | EndSeqNo | Y | — | SeqNum | Last message of range (inclusive). If the value of the `EndSeqNo` (#16) field is equal to the `BeginSeqNo` (#7) field, a single message is requested.<br>To request all messages after the `BeginSeqNo` (#7) field, set the `EndSeqNo` (#16) field to 0 (zero). |

For examples of this message, see "Examples" on page 50.

# 1.11 Session-level Reject (Bidirectional)

Both the provider system and the FX Grid send this message when a message is received but cannot be processed for some reason, such as missing a field required by the FIX protocol.

***Table 1-6** Reject Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 3 | String | 3=Reject |
| 45 | RefSeqNum | Y | — | SeqNum | Value of the `MsgSeqNum` (#34) field of the rejected message. |
| 58 | Text | N | — | String | Free-form text explaining the reason for rejection. |
| 371 | RefTagID | N | — | Int | The tag number of the FIX field being referenced. |
| 372 | RefMsgType | N | — | String | The value of the `MsgType` (#35) field in the FIX message being referenced. |
| 373 | SessionRejectReason | N | — | int | The code that indicates the reason for rejection.<br>0=Invalid tag number<br>1=Required tag missing<br>2=Tag not defined for message type<br>3=Undefined Tag<br>4=Tag specified without a value<br>5=Value is incorrect (out of range) for this tag<br>6=Incorrect data format for value<br>7=Decryption problem<br>8=Signature problem<br>9=CompID problem<br>10=SendingTime accuracy problem<br>11=Invalid MsgType<br>12=XML Validation error<br>13=Tag appears more than once<br>14=Tag specified out of required order<br>15=Repeating group fields out of order<br>16=Incorrect NumInGroup count for repeating group<br>17=Non "data" value includes field delimiter (SOH character)<br>99=Other |

For examples of this message, see "Examples" on page 50.

# 1.12 Sequence Reset (Bidirectional)

This message is sent by both the provider system and the FX Grid to reassign the sequence number to fill a gap in the messages that are being sent.

***Table 1-7** Sequence Reset Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 4 | String | 4=Sequence Reset |

***Table 1-7*** *Sequence Reset Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 122 | OrigSendingTime | N | — | UTCTimestamp | Original time of message transmission when transmitting orders as the result of a resend request |
| 36 | NewSeqNo | Y | — | SeqNum | The next sequence number to be transmitted. |
| 123 | GapFillFlag | N | — | Boolean | Indicates that the Sequence Reset message is replacing administrative or application messages that will not be resent.<br>Y=Gap fill, the `NewSeqNo` (#36) field should conform to the FIX-standard message sequencing rules.<br>N or not specified=Sequence reset, the `NewSeqNo` (#36) of the header of the Sequence Reset message should be ignored so that if it is not correct it does not generate resent requests. This value would be appropriate in a disaster recovery situation. |

For examples of this message, see "Examples" on page 50.

CHAPTER 2

# Trading Workflows

## 2.1 Introduction

All trading activity through the FIX Provider API begins with a query to establish the provider's trading session status. See "Starting the Trading Session" on page 22.

After the trading session has been established, the FIX Provider API implements the pricing workflow:

- "Executable Streaming Prices (ESP) Workflow" on page 24

For order submission and trade execution, see "Orders Workflow" on page 27.

## 2.2 Starting the Trading Session

To start a trading session, the following FIX message conversation must complete successfully between the FX Grid and the liquidity provider.

> **NOTE:** The FX Grid can only start the business workflow if the trading session status is "Open" (`TradSesStatus` (#340)=2) for both the price and order sessions.

1. Logon sent from provider on price session
2. Logon response on price session from Integral
3. Trading Session Status Request sent from Integral
4. Trading Session Status response with `TradSesStatus` (#340)=2 (Open) sent from provider on price session
5. Logon sent from provider on order session
6. Logon response on order session from Integral
7. Trading Session Status Request sent from Integral
8. Trading Session Status response with `TradSesStatus` (#340)=2 (Open) sent from provider on order session

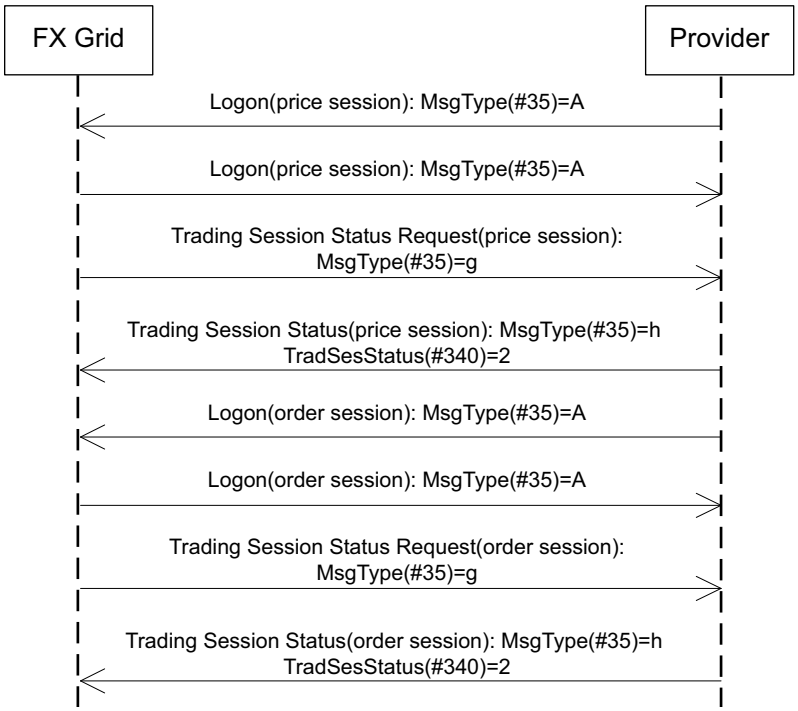


***Figure 2-1*** *Message Conversation To Start Trading*

## 2.3 Stopping the Trading Session

To alert customers of a change in trading session status, the liquidity provider sends an unsolicited trading session status message. Typical reasons for unsolicited trading session status messages are:

- End-of-day close for daily sequence number reset
- End-of-week close for weekly maintenance
- Intra-day close due to errors

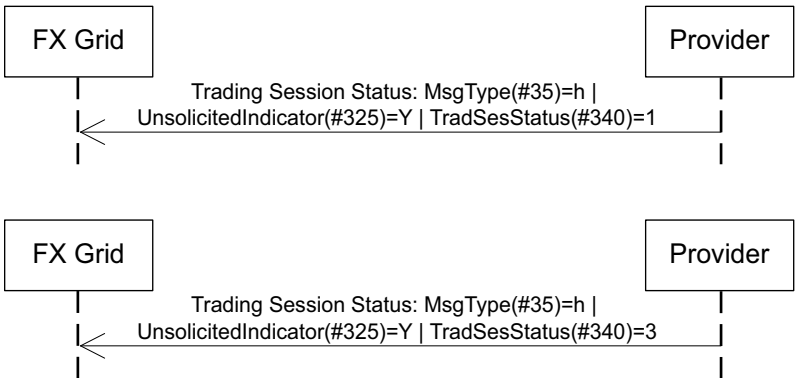


***Figure 2-2*** *Asynchronous Trading Session Status Update*

When the trading session is "Closed" (`TradSesStatus` (#340)=3) or "Halted" (`TradSesStatus` (#340)=1), the FX Grid stops sending messages and cleans up the current trading session. When the trading session is reopened, the FX Grid re-initializes the business workflow.

# 2.4 Executable Streaming Prices (ESP) Workflow

**NOTE:** The FX Grid can only start the trading workflow if the trading session status is "Open" (`TradSesStatus` (#340)=2) for both the price and order sessions. See "Starting the Trading Session" on page 22.

## 2.4.1 ESP Supported Trade Types

For the ESP workflow, the FIX Provider API currently supports trading in FX spot only.

## 2.4.2 ESP Workflow

The Executable Streaming Prices (ESP) workflow involves both price and order sessions. For more information about session types, see "Sessions" on page 14.

The customer requests market data in a price session, receives executable quotes on the same price session. Then in an order session the customer sends an order. For previously quoted orders (`OrdType` (#40)=D), the order refers to the originating price's `QuoteEntryID` (#299). For limit orders (`OrdType` (#40)=2), the order must include the requested price and size (`Price` (#44) and `OrderQty` (#38)). For market orders (`OrdType` (#40)=1), the order must include the requested size (`OrderQty` (#38))

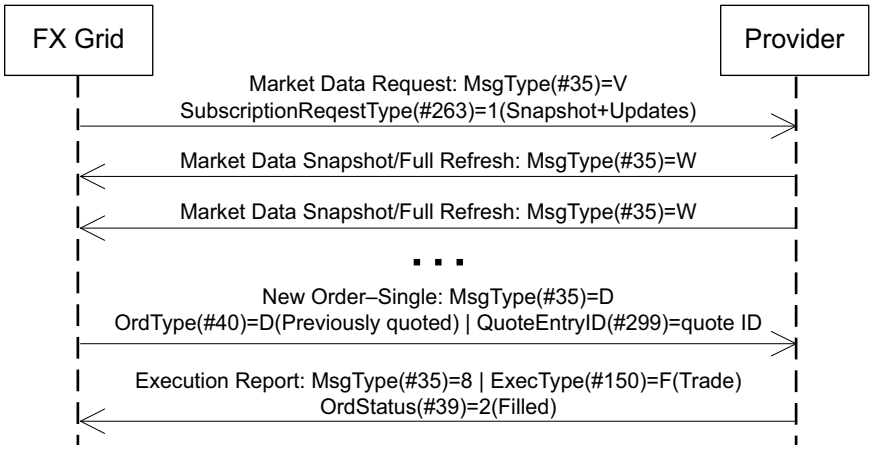


***Figure 2-3*** *ESP Trading Workflow: Previously Quoted Order*

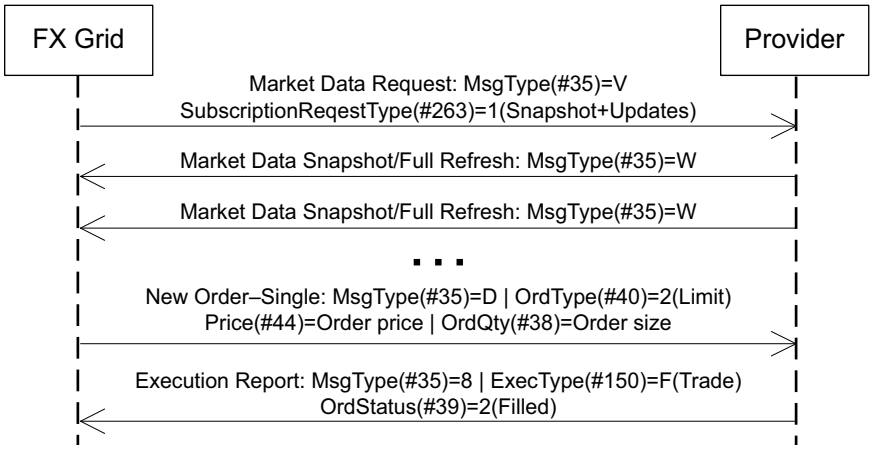


***Figure 2-4*** *ESP Trading Workflow: Limit Order*

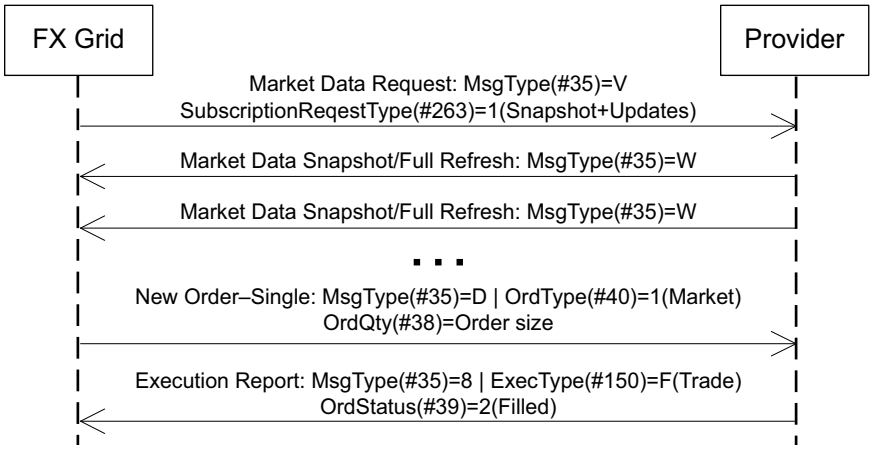


***Figure 2-5*** *ESP Trading Workflow: Market Order*

The following trading messages are applicable to the ESP workflow:

***Table 2-1*** *ESP Trading Messages*

| Message (Direction) | Session | Comments |
|---|---|---|
| "Market Data Request (FX Grid to Provider)" on page 35 | Price | Customer through FX Grid requests ESP from the liquidity provider or cancels ESP (unsubscribe) |
| "Market Data Snapshot/Full Refresh (Provider to FX Grid)" on page 36 | Price | Liquidity provider sends a price as a snapshot of the requested currency pairs to the customer through FX Grid |
| "Market Data Request Reject (Provider to FX Grid)" on page 38 | Price | Liquidity provider rejects the market data request (for example, the currency pair is not supported) |
| "New Order – Single (FX Grid to Provider)" on page 38 | Order | Customer through FX Grid submits an order to the liquidity provider in response to a price received in a Market Data Refresh message |

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

| Message (Direction) | Session | Comments |
|---|---|---|
| "Execution Report (Provider to FX Grid)" on page 42 | Order | Provider sends the current order status to the customer through FX Grid. See "Order State Transition" on page 32 for details. |

## 2.4.3 Quote Type

The Market Data Snapshot/Full Refresh message contains the latest market price for a currency pair. The FIX Provider API supports three types of ESP quotes: single price, multi-tiered, and multi-price.

### Single Price Quote

The Market Data Snapshot/Full Refresh contains one pair of bid and offer prices for a currency pair with each market data update.

- `NoMDEntries` (#268) = 2 (a single pair of bid and offer prices)
- `MDEntryType` (#269) = The message includes a bid entry and an offer entry.
- `MDEntryPositionNo` (#290) = 0

### Multi-Tiered Quote

The Market Data Snapshot/Full Refresh contains multi-tiered quotes for each amount level. For example, 1.2345/46 for amounts to 5M, 1.2344/47 for amounts to 10M, 1.2343/48 for amounts to 15M.

- `NoMDEntries` (#268) = 2 times the number of tiers. In the above example, the `NoMDEntries` (#268) value would be 6.
- `MDEntryType` (#269) = Typically the message includes an equal number of bid/offer prices. The number of bid and offer price can become uneven if the liquidity provider performs dynamic liquidity regeneration.
- `MDEntryPositionNo` (#290) = The value indicates the tier position of the bid and offer prices. In the first tier, prices have a value of 1. Second tier prices have a value of 2 and so on.

For previously quoted orders, the customer must refer to the quote ID in the `QuoteID` (#117) field of the New Order – Single message. For other order types (limit and market), the order price and size (`Price` (#44), `OrderQty` (#38)) must match the tier price and size of the quote in the `MDEntryPx` (#270) and `MDEntrySize` (#271) fields of the Market Data Snapshot/Full Refresh message. The liquidity provider should reject an order if the amount of the order and the price tier do not match. The order amount is filled to the offered amount.

### Multi-Price Quote

The Market Data Snapshot/Full Refresh contains the entire depth of the book with multiple independent prices (typically from an order book). A customer can sweep the entire book and get multiple fills at each price level.

- `NoMDEntries` (#268): Any number depending on the number of prices in the book
- `MDEntryType` (#269): The number of bid and offer prices is typically not even.
- `MDEntryPositionNo` (#290): 0 (each price is independent)

### 2.4.4 Quote Cancel

To remove or pull a price, you need to send a new Market Data Snapshot/Full Refresh message in the currency pair with the `QuoteCondition` (#276) field set to B (Closed / Inactive) or with a `MDEntrySize` (#271) value of 0 (zero). Do not send a Market Data Snapshot/Full Refresh message with a `MDEntryPx` (#270) value of 0 (zero).

### 2.4.5 Stream ID

Your organization can provide multiple streams and assign each stream to a customer category. You and Integral must agree on an ID for each stream.

> **NOTE:** The `QuoteID` (#117) value must be unique across all of your streams. It is your organization's responsibility to validate that the quote from a correct stream is used for a customer order.

The FX Grid sends the stream ID in the Market Data Request message in the `Stream ID` (#7540) custom field. The stream ID is optional. If your organization does not support multiple streams of prices, the `Stream ID` (#7540) field is left empty.

## 2.5 Orders Workflow

Order messages are exchanged on order sessions. See "Sessions" on page 14 for more information about order and price sessions. Orders originate with the FX Grid sending a New Order – Single message to the liquidity provider as part of a pricing workflow.

The following FIX messages implement the order trading flow:

***Table 2-2** Order Trading Messages*

| Message (Direction) | Session | Comments |
|---|---|---|
| "New Order – Single (FX Grid to Provider)" on page 38 | Order | Customer through the FX Grid submits an order to the provider. |
| "Execution Report (Provider to FX Grid)" on page 42 | Order | Provider sends the current order status to the customer through the FX Grid. The Execution Report with `ExecType` (#150) value 0 (New) is optional and may be skipped. See "Order State Transition" on page 32 for details. |
| "Order Timeout (FX Grid to Provider)" on page 41 | Order | If autocancellation is configured, provider receives this message if the FX Grid does not receive a rejection or trade verification before the order times out. |

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

| Message (Direction) | Session | Comments |
|---|---|---|
| "Don't Know Trade (DK) (FX Grid to Provider)" on page 41 | Order | If autocancellation is configured, provider receives this message if the FX Grid receives a trade response after the order times out and an Order Timeout message is sent. |

## 2.5.1 Order Submission and Execution

After the FX Grid submits an order, the provider sends an Execution Report message with an `ExecType` (#150) value of A="Pending New" to confirm receipt. After the provider has checked the order's validity, the provider sends an Execution Report message with an `ExecType` (#150) value of "New" to the FX Grid.

An order can have one or more trades executed against it, either one completely filling or multiple trades partially filling the order.

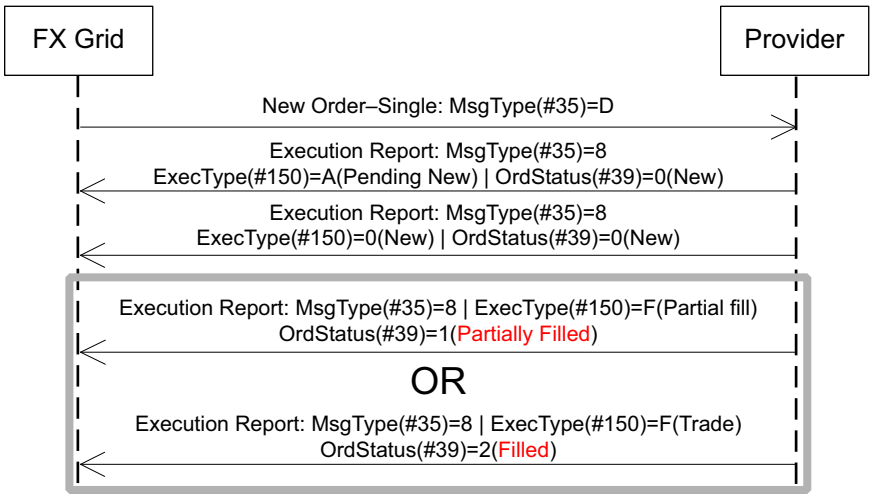


***Figure 2-6** Order Submission and Execution*

## 2.5.2 Order Rejection

After the FX Grid submits an order, the provider immediately sends an Execution Message with an `ExecType` (#150) value of A="Pending New" to confirm receipt. If the order fails the provider's validity check, the provider sends an Execution Report message with an `ExecType` (#150) value of "Rejected" to the FX Grid.

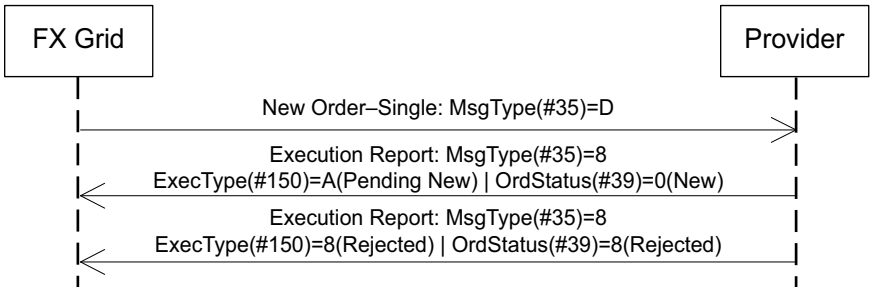


***Figure 2-7*** *Order Rejection*

## 2.5.3 Order Expiry

An IOC order expires immediately after the initial match. Any residual amount is canceled by the provider by sending an Execution Report message with an `ExecType` (#150) value of "Expired" to the FX Grid.

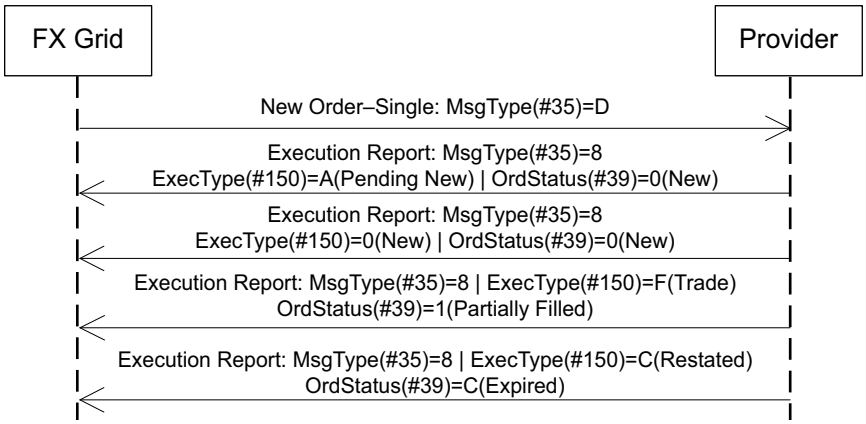


***Figure 2-8*** *Order Expiry*

## 2.5.4 Order Timeout

When autocancellation is enabled, if the FX Grid does not receive a trade rejection or verification from the liquidity provider by the end of the timeout interval, an Order Timeout message is sent to the provider. This order timeout informs the provider that no valid Execution Report with a fill or rejection was received within the autocancellation timeout interval and the order is timed out (effectively rejected).

If a rejection or verification is received from the provider after the Order Timeout message is sent, a Don't Know Trade (DK) message is sent to the provider.

In addition, when an order reaches a terminal state (verification, rejection, expiry for a partial fill, or expiry/complete fill for a multiple fill order), then any terminal state message that is sent by the provider triggers a Don't Know Trade (DK) message in response.

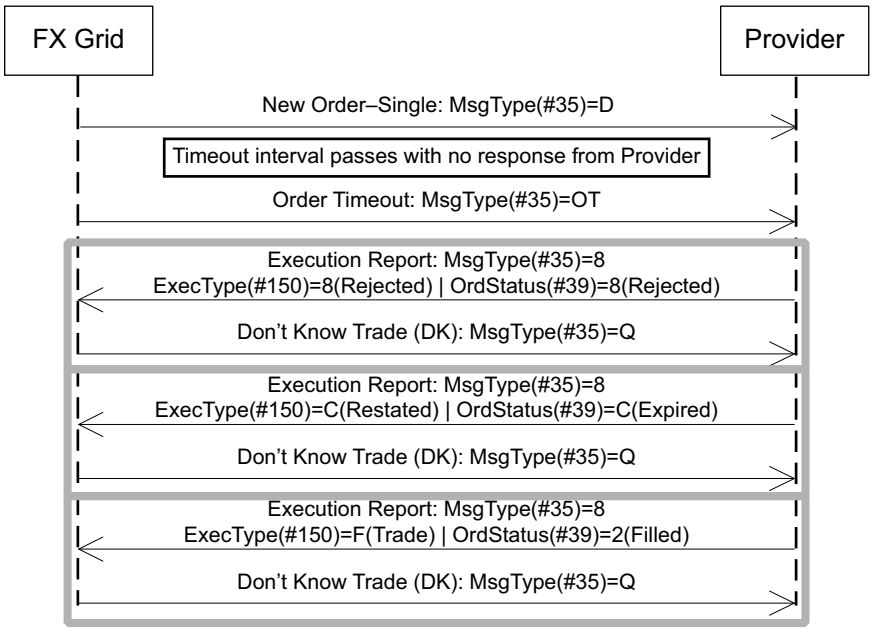


***Figure 2-9*** *Order Timeout*

## 2.5.5 Supported Order Types

The FIX Provider API supports orders with the following `OrdType` (#40) values. The FX Grid rejects messages from customers with unsupported order types.

***Table 2-3*** *Order Types*

| Order Type | OrdType (#40) Value | Description |
|---|---|---|
| Previously Quoted | D | The client sends new orders with a reference to a previously received executable price in `QuoteEntryID` (#299) from a price ("Market Data Snapshot/Full Refresh (Provider to FX Grid)" on page 36). |
| Limit | 2 | Orders executed at the limit price or better until they are filled, rejected, or expired ("Order Expiry" on page 30). Orders must include the requested price and size (`Price` (#44) and `OrderQty` (#38)). |
| Market | 1 | Orders are executed at the best available price in the system.until they are filled, rejected, or expired ("Order Expiry" on page 30). |

## 2.5.6 Order Expiry

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

***Table 2-4** Order Expiry*

| 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. If the order is not completely filled (or after the expiry/cancel message is received) any remaining amount is cancelled. If no fill/expiry/cancel message is received, the entire order is assumed to be cancelled.<br>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, otherwise the order goes into a pending state. Subsequent fills, if any, may be of any size.<br>If `MinQty` (#110) is not specified, the first fill may be of any size less than or equal to the order size. Subsequent fills, if any, may be of any size. | 3=IOC | ■ Market<br>■ Limit<br>■ 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. When this happens, all previous fills, if any, are ignored and the order is cancelled. | 4=FOK | ■ Market<br>■ Limit<br>■ Previously Quoted |
| Good Till Date/Time (GTD) | The order remains active until one of following conditions is met:<br>■ Fully filled<br>■ Expired<br>For GTD orders, the order expiry time must be specified in GMT in the `ExpireTime` (#126) field. Applicable to limit orders only. See "Orders Workflow" on page 27. | 6=GTD | ■ Market<br>■ Limit<br>■ Previously Quoted |

## 2.5.7 Partial Fills

**NOTE:** Partial fills apply only to the order trading workflow. See "Orders Workflow" on page 27 for information.

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

- Partial fill: The order amount can be filled multiple times until the entire amount is filled.
- Partial fill with market minimum: The order amount can be filled multiple times, but the first fill must be no less than the market minimum defined by the `MinQty` (#110) field. Subsequent fills have no size requirement.
- No partial fill: The order amount must be filled in its entirety with exactly one fill. The value of the `MinQty` (#110) field equals the value of the `OrderQty` (#38) field. This is effectively a fill-or-kill order.

For more details about partial fills, see the "MinQty" field in the "New Order – Single (FX Grid to Provider)" message on page 38.

## 2.5.8 Order State Transition

The provider maintains an order's status and propagates changes in status via Execution Report messages to the FX Grid. The order states and state transitions are shown in Figure 2-10 on page 32. Each circle represents a value of the `OrdStatus` (#39) field of an Execution Report message. The event names on the transition lines represent the `ExecType` (#150) values of the Execution Report message. Multiple transitions from the same `OrdStatus` (#39) may have the same `ExecType` (#150) value. The destination `OrdStatus` (#39) is determined by other attributes of the order as described in "Orders Workflow" on page 27.

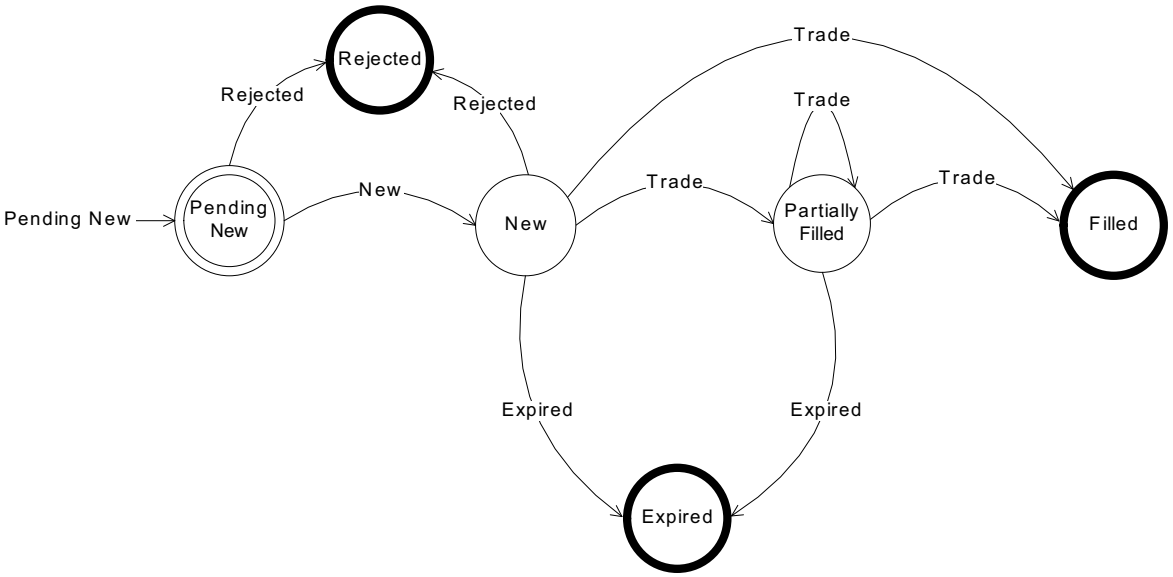


***Figure 2-10*** *Order State Transitions*

***Table 2-5*** *Order Status*

| Order Status | OrdStatus (#39) | Description |
|---|---|---|
| Pending New | A | An order is received by liquidity provider. This order state is optional. |
| New | 0 (zero) | The validity of the order is confirmed. The order is successfully entered liquidity provider's execution management process. |
| Partially filled | 1 | The order is partially filled by liquidity provider. |
| Filled | 2 | The total order amount is filled. |
| Rejected | 8 | The provider determines that the order is invalid. The order is rejected. |
| Expired | C | When an order is expired, its residual amount (full amount if the order is New) is canceled. For IOC orders, the order is immediately expired after the initial match. Any remaining amount must be expired or rejected. |

**NOTE:** `OrdStatus` (#39)=4 (Canceled) is not supported in the FIX Provider API.

CHAPTER 3

# Trading Messages

## 3.1 Introduction

The FIX Provider API offers the following order workflow features to liquidity providers:

- Order submission
- Order execution reports for accept/acknowledge, reject, trade execution, and partial fill events

## 3.2 Supported Message Types

The FIX Provider API order workflow supports the following FIX messages types:

***Table 3-1*** *Supported Message Types*

| Message Type—MsgType (#35) Field Value | Message Name | Inbound to Provider (I), Outbound from Provider (O), or Bidirectional (B) |
|---|---|---|
| g | Trading Session Status Request on page 34 | I |
| h | Trading Session Status on page 34 | O |
| V | Market Data Request on page 35 | I |
| W | Market Data Snapshot/Full Refresh on page 36 | O |
| Y | Market Data Request Reject on page 38 | O |
| D | New Order – Single on page 38 | I |
| OT | Order Timeout on page 41 | I |
| Q | Don't Know Trade on page 41 | I |
| 8 | Execution Report on page 42 | O |
| j | Business Message Reject on page 45 | B |

# 3.3 Trading Session

## 3.3.1 Trading Session Status Request (FX Grid to Provider)

This message is sent by the FX Grid to query the provider's status and as part of starting a trading session with the provider. For more information, see "Starting the Trading Session" on page 22.

***Table 3-2** Trading Session Status Request Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | g | String | g=Trading Session Status Request |
| 263 | SubscriptionRequestType | Y | — | char | 0 (zero)=Snapshot |
| 335 | TradSesReqID | Y | — | String | Unique ID assigned by the customer for the request message |

For examples of this message, see "Examples" on page 50.

## 3.3.2 Trading Session Status (Provider to FX Grid)

This message is sent by the provider to notify the FX Grid of intra-day closing and opening of the market, and as part of starting a trading session with the provider. For more information, see "Starting the Trading Session" on page 22.

***Table 3-3** Trading Session Status Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Comments |
|---|---|---|---|---|---|
| 35 | MsgType | Y | h | String | h=Trading Session Status |
| 325 | UnsolicitedIndicator | N | ■ Y<br>■ N | Boolean | If the message is sent unsolicited, the value of this field should be "Y". If the message is sent as a response to a customer request, this field should have a value of "N" or may be omitted or left empty. For more information, see "Stopping the Trading Session" on page 23. |
| 335 | TradSesReqID | N | — | String | Reference to the value of `TradSesReqID` (#335) field on the customer request, if any. If the status message is sent unsolicited, this field should be left empty or not included. |
| 336 | TradingSessionID | Y | — | String | ID of the trading session. Can be any string. |
| 340 | TradSesStatus | Y | — | int | 0=Unknown<br>1=Halted (trading temporarily suspended)<br>2=Open<br>3=Closed |

For examples of this message, see "Examples" on page 50.

# 3.4 Market Data Messages

The messages in this section are used to access the ESP workflow (see "Executable Streaming Prices (ESP) Workflow" on page 24).

## 3.4.1 Market Data Request (FX Grid to Provider)

The FX Grid sends one Market Data Request message per currency pair per price stream (across all customers) to initiate trading in the ESP workflow with the liquidity provider. If a customer subscribes to five currency pairs in a price stream, the provider receives five Market Data Request messages. For two price streams with five unique currency pairs each across multiple customers, the provider receives ten Market Data Request messages. The subscription is sent when the first customer set up for a currency pair on the stream subscribes. Because a subscription is already sent, when a subsequent customer requests prices for the same currency pair on the stream, no further subscription is sent to the liquidity provider.

The FX Grid also sends a single Market Data Request message on behalf of a customer to unsubscribe from all rates. The unsubuscription message is sent per currency pair per price stream and is sent only when the last customer receiving pricing for the currency pair per stream no longer needs it. See "Executable Streaming Prices (ESP) Workflow" on page 24 for information.

***Table 3-4*** *Market Data Request Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | V | String | V=Market Data Request |
| 262 | MDReqID | Y | — | String | The customer-assigned unique ID for the market data request. This ID must not contain the ampersand character "@". The FX Grid rejects all requests with duplicate IDs. |
| 263 | SubscriptionRequestType | Y | 1=Snapshot + Updates (subscribe to stream, default)<br>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 | int | Depth of market for book snapshot. The only valid value for this field in the FIX Provider API is 0=Full book. |
| 265 | MDUpdateType | Y | 0=Full Refresh | int | This value specifies the type of Market Data update. This field is required with the `SubscriptionRequestType` (#263) value of `1` (Snapshot + Updates). |
| 7540 | StreamID | N | — | String | The ID for the stream agreed by Integral and the liquidity provider.<br>If the liquidity provider does not support multi-stream pricing, this field is not included or is left empty. |
| **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.** | | | | | |

***Table 3-4** Market Data Request Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 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 variable currencies of the currency pair in the following format:<br>*baseCCY*/*variableCCY*<br>(for example, "EUR/USD") |
| 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 |
| **End of repeating group** | | | | | |
| **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.)<br>■ 0 (zero)<br>■ 1 (one) | char | The FIX Provider API supports only two-way market data. This field is required by the FIX specification but is ignored by the FIX Provider 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** | | | | | |

For examples of this message, see "Examples" on page 50.

## 3.4.2 Market Data Snapshot/Full Refresh (Provider to FX Grid)

The provider sends a Market Data Snapshot/Full Refresh message to the FX Grid in response to a Market Data Request message for each currency pair stream.

The FIX Provider API maintains a one-to-one relationship between customers, currency pairs, and Market Data Snapshot/Full Refresh message targets. The FX Grid sends Market Data Snapshot/Full Refresh messages for each customer and each currency pair stream in the ESP workflow.

The Market Data Snapshot/Full Refresh message may contain multiple rates. The `NoMDEntries` (#268) field indicates the number of rates in the message. If you want to send multi-tier or multi-price quotes, see "Quote Type" on page 26. The customer receives the fully qualified customer rate and associated limits. The customer rate is specified by the `MDEntryPx` (#270) field.

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

***Table 3-5*** *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 |
| 55 | Symbol | Y | — | String | The symbol for the base and variable currencies of the currency pair in the following format:<br>*baseCurrency/termCurrency*<br>(for example, "EUR/USD")<br>For inverted quotes, this field should be specified as:<br>*termCurrency/baseCurrency*<br>(for example, "USD/EUR") |
| 262 | MDReqID | Y | — | String | Unique identifier from the originating Market Data Request |
| 460 | Product | N | 4 | int | The asset class. The value is always 4=CURRENCY. |
| 541 | MaturityDate | N | — | LocalMktDate | You should leave this field empty and specify `SettlDate` (#64) instead. The value of the MaturityDate (#541) field is populated by the FX Grid. Date of maturity in *YYYYMMDD* format. |
| **The following shaded rows are a repeating group of fields that represent one currency pair. The required fields must be included as a group for each currency pair or your request will be rejected. The value of the NoMDEntries (#268) field indicates the number of groups and thus the number of currency pairs.** | | | | | |
| 268 | NoMDEntries | Y | — | NumInGroup | Number of entries in market data message. Each bid and offer represents one market data entry. If three bid/offer dealing prices are included, the value of the `NoMDEntries` (#268) field is `6`. |
| 269 | MDEntryType | Y | — | char | The side of the rate. Part of the repeating group of fields for each rate in the update.<br>0=Bid<br>1=Offer<br>2=Trade |
| 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. The value of this field should not be set to 0 (zero). Part of the repeating group of fields for each rate in the update. |
| 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. Part of the repeating group of fields for each rate in the update. |
| 271 | MDEntrySize | Y | — | Qty | The quantity (in the case of multiple tiers, the limit). Part of the repeating group of fields for each rate in the update. |
| 276 | QuoteCondition | Y | — | MultipleValueString | Indicates whether the rate is tradable or only indicative. Part of the repeating group of fields for each rate in the update.<br>A=Open/Active<br>B=Closed/Inactive |
| 282 | MDEntryOriginator | Y | — | String | The provider organization ID. The provider associated with the bid or offer quote. Part of the repeating group of fields for each rate in the update. This field is not used in Integral processing. |
| 299 | QuoteEntryID | Y | — | String | Uniquely identifies each rate as part of a quote set. The reference ID for the dealing price. Part of the repeating group of fields for each rate in the update. |

***Table 3-5*** *Market Data Snapshot/Full Refresh Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 290 | MDEntryPositionNo | Y | — | int | The integer value indicates the tier of the price. For multi-tier quotes, the value will be greater than zero. For single price or multi-price quotes, this field equals zero. See "Quote Type" on page 26 for information about business rules regarding multi-price and multi-tier quotes. |
| **End of repeating group** | | | | | |

For examples of this message for both multi-price and multi-tier quotes, see "Examples" on page 50.

### 3.4.3 Market Data Request Reject (Provider to FX Grid)

The liquidity provider sends a Market Data Request Reject message to the FX Grid in response to an unsuccessful Market Data Request message.

***Table 3-6*** *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 |
| 58 | Text | N | — | String | Your organization's description of the rejection should be sent in this field. |
| 262 | MDReqID | Y | — | String | The unique identifier from the originating Market Data Request. |
| 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.<br>0=Unknown symbol<br>1=Duplicate `MDReqID`<br>3=Insufficient Permissions |

For examples of this message, see "Examples" on page 50.

## 3.5 Trading Messages

The messages in this section are used to access the trading workflow of the FIX Provider API for all trading workflows (see "Trading Workflows" on page 22).

### 3.5.1 New Order – Single (FX Grid to Provider)

The FX Grid sends a New Order – Single message to submit an order to the liquidity provider.

The order is an execution request on a previously quoted price (`OrdType` (#40)=D).

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

***Table 3-7*** *New Order – Single Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | D | String | D=New Order – Single |
| 1 | Account | Y | — | String | The account ID. The value of this field is agreed upon by your organization and Integral. |
| 11 | ClOrdID | Y | — | String | A session-scoped unique order ID assigned by the FX Grid. |
| 15 | Currency | Y | — | Currency | The dealt currency. This may be the base or term currency of a currency pair. |
| 21 | HandlInst | Y | 1 (one) | char | Instructions for order handling on the broker trading floor.<br>The value of this field is always:<br>1 = Automated execution order, private, no Broker intervention |
| 38 | OrderQty | Y | — | Qty | FX spot: 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). |
| 40 | OrdType | Y | D=Previously quoted<br>2=Limit<br>1=Market | char | Other order types are not currently supported. See "Supported Order Types" on page 30. |
| 44 | Price | Y | — | Price | 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. |
| 54 | Side | Y | 1=Buy (Bid)<br>2=Sell (Offer) | char | FX spot: The side of the order from the customer's perspective and in terms of the dealt currency `Currency` (#15). |
| 55 | Symbol | Y | — | String | The symbol for the base and variable currencies of the currency pair in the following format:<br>*baseCurrency/termCurrency*<br>(for example, "EUR/USD")<br>For inverted quotes, this field should be specified as:<br>*termCurrency/baseCurrency*<br>(for example, "USD/EUR") |
| 58 | Text | N | — | String | Free format text string |
| 59 | TimeInForce | Y | ■ 3=IOC<br>■ 4=FOK<br>■ 6=GTD | char | See "Order Expiry" on page 30 for a description of order expiry types. |
| 60 | TransactTime | Y | — | UTCTimestamp | Time automatically stamped by the server that this order request was initiated/released by the trader, trading system, or intermediary in the format:<br>*YYYYMMDD-HH:MM:SS* |

***Table 3-7*** *New Order – Single Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 64 | FutSettDate | Y | ■ Spot date or tenor | LocalMktDate | ■ FX spot: "SPOT" tenor or spot date in *YYYYMMDD* format. |
| 110 | MinQty | N | ■ 0 (zero)<br>■ Less than `OrderQty` (#38)<br>■ Equal to `OrderQty` (#38) | Qty | This field is sent only if the incoming customer order specifies a minimum quantity. This field can be ignored if `TimeInForce` (#59) is 4 (FOK).<br>Specifies how the order can be filled:<br>■ Partial fill: If the value is 0 (zero), the order amount can be filled multiple times until the entire amount is filled.<br>■ Partial fill with market minimum: The order amount can be filled multiple times, but the first fill must be no less than the market minimum defined by the `MinQty` (#110) field. Subsequent fills have no size requirement.<br>■ 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 one fill. This is effectively an FOK order. |
| 117 | QuoteID | See descr. | — | String | This is only required for `OrdType` (#40) =`D` (previously quoted) orders. The reference ID of the bid or offer dealing price. |
| 167 | SecurityType | Y | FOR | String | FOR=Foreign Exchange Contract |
| 168 | EffectiveTime | N | — | UTCTimestamp | The order submission time in GMT in the format:<br>*YYYYMMDD-HH:MM:SS* |
| 7540 | StreamID | N | — | String | The ID for the stream agreed by Integral and the liquidity provider.<br>If the liquidity provider does not support multi-stream pricing, this field is not included or is left empty. |
| **The following shaded rows consist of the Parties component, a group of fields that represent the ClientTag on the trade request, if any. If this component is included in the message, the value of the NoPartyIDs (#453) field always has the value of 1 (one), indicating a ClientTag.** | | | | | |
| 453 | NoPartyIDs | N | 1 (one) | 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 ClientTag on the trade request. |
| 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** | | | | | |

For examples of this message, see "Examples" on page 50.

## 3.5.2 Order Timeout (FX Grid to Provider)

When the FX Grid is configured to automatically cancel timed-out orders, this message is sent to inform the provider that the FX Grid did not receive an Execution Report message response (either an order rejection or a trade verification) for the order within the time-out period. The original order is effectively rejected. If the FX Grid receives a trade verification or rejection after the order has been timed out, the FX Grid sends a Don't Know Trade message to the provider. This order time-out conversation happens in an order session. See "Order Timeout" on page 29 for more information about the order timeout workflow.

***Table 3-8*** *Order Timeout Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | OT | String | OT=Order Timeout |
| 11 | ClOrdID | Y | — | String | The unique ID assigned by the FX Grid to the order, the `ClOrdID` (#11) of the originating New Order – Single message. This is the order for which no response was received. |

For examples of this message, see "Examples" on page 50.

## 3.5.3 Don't Know Trade (DK) (FX Grid to Provider)

The Don't Know Trade (DK) message is sent to the liquidity provider if a fill, rejection, or any other terminal-state message is received by the FX Grid for an order that has been timed out. See "Order Timeout" on page 29 for more information about the order timeout workflow.

***Table 3-9*** *Don't Know Trade (DK) Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | Q | String | Q=Don't Know Trade (DK) |
| 17 | ExecID | Y | — | String | The provider's unique trade ID of problem execution |
| 54 | Side | See descr. | — | char | Required by the FIX specification, but not required by the FIX Provider API. This field is included if it is present in the Execution Report from the maker. .The trade's side from customer's perspective. For FX swap, it is the side of the far leg.<br>1=Buy (Bid)<br>2=Sell (Offer) |
| 37 | OrderID | Y | — | String | The provider's order ID on the problem execution |
| 55 | Symbol | Y | — | String | The symbol for the base and variable currencies of the currency pair in the following format:<br>*baseCCY/variableCCY*<br>(for example, "EUR/USD") |
| 127 | DKReason | Y | D | char | Reason for execution rejection.<br>D=No matching order |

## 3.5.4 Execution Report (Provider to FX Grid)

The liquidity provider sends an Execution Report message for the following events:

- Order filled/partially filled

The combination of the `OrdStatus` (#39) and `ExecType` (#150) fields indicate the current state of the order. See "Order State Transition" on page 32 for more information about order status.

***Table 3-10** Execution Report Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | 8 | String | 8=Execution Report |
| 37 | OrderID | Y | — | String | The unique order ID assigned by your organization. |
| 11 | ClOrdID | Y | — | String | A unique order ID assigned by the FX Grid. It is the same value as sent by the customer in the New Order - Single message. |
| 17 | ExecID | Y | — | String | Unique ID for each order execution report. |
| 40 | OrdType | N | D=Previously quoted<br>2=Limit<br>1=Market | char | The order type. See "Supported Order Types" on page 30. |
| 150 | ExecType | Y | — | char | Describes the type of execution report. These are the state transition events illustrated in Figure 2-10 on page 32 except Order Status:<br>■ A=Pending new<br>■ 0=New<br>■ 8=Rejected<br>■ C=Expired<br>■ F=Trade<br>■ I=Order Status<br>■ 2=Fill |
| 39 | OrdStatus | Y | — | char | Describes the current state of the order. The value 4=Canceled is not applicable to the FIX Provider API.<br>Valid values:<br>■ A=Pending New<br>■ 0=New (outstanding)<br>■ 1=Partial Fill (after order matching)<br>■ 2=Filled (after order matching)<br>■ 8=Rejected (before or after order matching)<br>■ C=Expired |

***Table 3-10*** *Execution Report Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 103 | OrdRejReason | N | — | int | Code to identify reason for order rejection.<br>Valid values:<br>■ 0=Broker / Exchange option<br>■ 1=Unknown symbol<br>■ 2=Exchange closed<br>■ 3=Order exceeds limit<br>■ 4=Too late to enter<br>■ 5=Unknown Order<br>■ 6=Duplicate Order (duplicate `ClOrdID` (#11))<br>■ 7=Duplicate of a verbally communicated order<br>■ 8=Stale Order<br>■ 9=Trade Along required<br>■ 10=Invalid Investor ID<br>■ 11=Unsupported order characteristic<br>■ 12=Surveillance Option |
| 1 | Account | Y | — | String | The account ID. The value of this field is agreed upon by your organization and Integral. |
| 32 | LastQty | See descr. | — | Qty | For fills or partial fills, the quantity bought or sold on this fill.<br>If a fill or partial fill is received with this field not specified or set to zero, the FX Grid rejects it with a Business Message Reject message. |
| 119 | SettlCurrAmt | Y | — | Amt | FX spot: Settled amount in terms of settlement currency specified by the `SettlCurrency` (#120) field. |
| 120 | SettlCurrency | Y | — | Currency | FX spot: Settled currency |
| 55 | Symbol | Y | — | String | The symbol for the base and variable currencies of the currency pair in the following format:<br>*baseCCY/variableCCY*<br>(for example, "EUR/USD")<br>This should be the same value as the one received from the associated New Order - Single message. |
| 54 | Side | Y | — | char | The order side from customer's perspective. For FX swap, it is the side of the far leg.<br>1=Buy (Bid)<br>2=Sell (Offer)<br>This should be the same value as the one received from the associated New Order - Single message. |
| 38 | OrderQty | N | — | Qty | FX spot: The total quantity ordered |

***Table 3-10*** *Execution Report Message Fields (continued)*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 44 | Price | N | — | Qty | Do not use this field to determine the price of a fill. Use `LastPx` (#31) instead. This should be the same value as the one received from the associated New Order - Single message.<br>FX spot: The execution price for previously quoted orders. The precision of this float value must be agreed upon by both the sending and the receiving parties. |
| 15 | Currency | N | — | Currency | The dealt currency. This may be the base or term currency of a currency pair. |
| 167 | SecurityType | Y | FOR | String | FOR=Foreign Exchange Contract |
| 60 | TransactTime | Y | — | UTCTimestamp | The timestamp of the Execution Report creation.<br>The format is:<br>*YYYYMMDD-HH:MM:SS* |
| 59 | TimeInForce | N | 3=IOC<br>4=FOK<br>6=GTD | char | The time-in-force of the order. See "Order Expiry" on page 29. |
| 168 | EffectiveTime | N | — | UTCTimestamp | The order submission time in GMT in the format:<br>*YYYYMMDD-HH:MM:SS* |
| 151 | LeavesQty | Y | — | Qty | Amount not filled (equals `OrderQty` (#38) minus `CumQty` (#14)<br>FX spot: Open amount. Responses for a complete fill or rejection are "0" (zero). Any value other than zero indicates a partial fill. |
| 14 | CumQty | Y | — | Qty | FX spot: Total filled amount. If `OrdStatus` (#39) =F (partial fill), this value is total amount filled at the time of the current fill. If `OrdStatus` (#39)=2 (filled), the value of this field is the same as the `OrderQty` (#38) field. If `OrdStatus` (#39)=8 (rejected), the value of this field is "0" (zero). |
| 31 | LastPx | See descr. | — | Price | The price of a fill to ensure trade reconciliation and to determine any price improvement.<br>FX spot: Price at which the current or last fill was made. |
| 32 | LastQty | N | — | Qty | Amount for this or the last fill. |
| 6 | AvgPx | Y | — | Price | The average execution price for the filled amount specified in `CumQty` (#14) field). If this field is set, then the `CumQty` (#14) field must be set to a non-zero value to avoid errors. The precision of this float value must be agreed upon by both the sending and the receiving parties. Do not use this field to determine the price of a fill. Use `LastPx` (#31) instead. |

***Table 3-10** Execution Report Message 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 Order Execution Report message is sent because of order rejection (the `ExecType` (#150) field="8").<br>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. |
| 75 | TradeDate | N | — | LocalMktDate | The trade date (*YYYYMMDD*) |

For examples of this message, see "Examples" on page 50.

## 3.5.5 Business Message Reject (Bidirectional)

The Business Message Reject message is sent by both the FX Grid and the liquidity provider:

- The FX Grid sends the Business Message Reject message as part of the RFS workflow to cancel a Quote Request (`RefMsgType` (#372)=`Z`). See "Trading Workflows" on page 22.
- For Fixed Spot Rolls (FSRs), if the quote rates do not match the customer-specified rate, the quotes are dropped and a Business Message Reject is sent to cancel the quote message. See "Trading Messages" on page 33.
- If an Execution Report message for a fill or partial fill is received with the `LastQty` (#32) field not specified or set to zero, the FX Grid rejects it with a Business Message Reject message. See "LastQty" on page 43.

***Table 3-11** Business Message Reject Message Fields*

| Tag | Field Name | Req'd | Value | FIX Format | Description |
|---|---|---|---|---|---|
| 35 | MsgType | Y | j | String | j=Business Message Reject |
| 372 | RefMsgType | Y | — | String | The `MsgType` (#35) of the message referred to:<br>Z=Quote Cancel (from client to cancel a Quote Request) |
| 379 | BusinessRejectRefID | N | — | String | If the message is sent to cancel a quote request (`RefMsgType` (#372) =Z), this field contains the `QuoteID` (#117) of the quote being canceled. |
| 58 | Text | N | — | String | Free-format text string |

APPENDIX A

# Changes

## A.1 Changes

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

***Table A-1*** *Document Changes*

| Date | API Version | Enhancements |
|---|---|---|
| April 2015 (2.1v2) | 2.0 | ■ The required status of `LastSpotRate` (#194) and `LastForwardPoints` (#195) on the Execution Report message has been corrected and clarified. See "Execution Report (Provider to FX Grid)" on page 42. |
| June 2014 (2.0v7) | 2.0 | ■ You can now quote FSRs (Fixed Spot Rolls) via the FIX Provider API. See:<br>□ "FIX Solution" in "Supported Deal Types" on page 16<br>□ "Trading Messages" on page 33<br>□ "Trading Messages" on page 33<br>□ New example files: `FSR_QuoteRequest.txt`, `FSR_Quote.txt`, `FSR_NewOrderSingle.txt`, `FSR_ExecutionReport_PendingNew.txt`, `FSR_ExecutionReport_New.txt`, `FSR_ExecutionReport_Trade.txt` |
| November 2013 (2.0v6) | 2.0 | ■ Corrected description of Market Data Request message ("Market Data Request (FX Grid to Provider)" on page 35).<br>■ Removed references to Order Status Request message. This message type is not currently supported by the FIX Provider API.<br>■ Enhanced description of Business Message Reject message ("Business Message Reject (Bidirectional)" on page 45). |

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

| Date | API Version | Enhancements |
|---|---|---|
| July 2013 (2.0v5) | 2.0 | ■ Added support for NDF trades:<br>□ "Supported Deal Types" on page 16<br>□ Quote Request: "Trading Messages" on page 33<br>□ Quote Request: "Trading Messages" on page 33<br>□ Quote Request: "Trading Messages" on page 33<br>□ Quote: "Trading Messages" on page 33<br>□ Quote: "Trading Messages" on page 33<br>□ New Order – Single: "OrdType" on page 39<br>□ New Order – Single: "Trading Messages" on page 33<br>□ New Order – Single: "For examples of this message, see "Examples" on page 50." on page 41<br>□ Execution Report: "Trading Messages" on page 33<br>□ Execution Report: "For examples of this message, see "Examples" on page 50." on page 45<br>■ Added support for FSR trades:<br>□ "Supported Deal Types" on page 16<br>□ Quote Request: "Trading Messages" on page 33<br>■ Added mid-mark price information:<br>□ "Mid Mark Price" on page 16<br>□ Quote: "Trading Messages" on page 33 and "Trading Messages" on page 33<br>■ The descriptions of the IOC and FOK times in force have been expanded. See "Immediate or Cancel (IOC)" on page 31 and "Fill or Kill (FOK)" on page 31.<br>■ The `MinQty` (#110) field on an order affects how IOC orders must be filled. See "Immediate or Cancel (IOC)" on page 31. |
| January 2012 (1.7.3v1) | 1.7.3 | ■ Added `Stream ID` (#7540) to "Trading Messages" on page 33, "Trading Messages" on page 33, "New Order – Single (FX Grid to Provider)" on page 38.<br>■ Added `Side` (#54) to "Don't Know Trade (DK) (FX Grid to Provider)" on page 41.<br>■ Changed the required status of the following fields:<br>□ `MinQty` (#110) field changed to "N" on "New Order – Single (FX Grid to Provider)" on page 38. The field is sent only if the incoming customer order has `MinQty` (#110) set. |
| October 2011 (1.6v3) | 1.6 | ■ Clarified one-to-one relationship between customers, currency pairs, and Market Data Request messages in description of "Market Data Request (FX Grid to Provider)" on page 35. |

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

| Date | API Version | Enhancements |
|---|---|---|
| September 2011 (1.6v2) | 1.6 | ■ Changed "SubAccountID" to "ClientTag" and expanded definition of ClientTag. See "Account IDs" on page 12.<br>■ Corrected ID formats in "Liquidity Provider Organization ID" on page 13 and "FX Grid Server ID" on page 13.<br>■ Description of connection modes enhanced to clarify acceptor mode as the default and initiator as the special case that requires coordination with Integral. See "Connection Mode" on page 13.<br>■ Enhanced section on start/stop times with examples and format for custom start/stop times. See "Business Day End and Start" on page 15.<br>■ Corrected list of supported deal types by workflow. See "Supported Deal Types" on page 16.<br>■ Clarified requirement for sequence number reset on log-on. Added information about heartbeat interval. See "Logon (Bidirectional)" on page 17.<br>■ Clarified that Resend Request messages are valid only on order sessions. See "Resend Request (Bidirectional)" on page 19.<br>■ The chapter "Trading Workflows" on page 22 has been reworked to describe two pricing workflows (EPS/RFS) and the resulting order workflow.<br>■ Added limit order example to "ESP Workflow" on page 24. |
| September 2011 (1.6v2) continued | 1.6 | ■ The Execution Report message with `ExecType` (#150) value of A="Pending New" is optional in "Order Submission and Execution" on page 28.<br>■ The order state `OrdStatus` (#39)=4 (Canceled) is not applicable to the FIX Provider API. See "Execution Report (Provider to FX Grid)" on page 42. |
| July 2011 | 1.6 | ■ The section "Trading Session Status" has been expanded and divided into two new sections. See "Starting the Trading Session" on page 22 and "Stopping the Trading Session" on page 23.<br>■ Changed all references to "quote" session to "price" session, including naming convention for liquidity provider ID. See "Liquidity Provider Organization ID" on page 13.<br>■ Changed required status of `Symbol` (#55) from "N" to "Y" in "Execution Report (Provider to FX Grid)" on page 42.<br>■ Examples have been updated and moved into separate text files for all messages.<br>■ Added order timeout workflow:<br>□ "Orders Workflow" on page 27<br>□ "Order Timeout" on page 29<br>□ "Order Timeout (FX Grid to Provider)" on page 41<br>□ "Don't Know Trade (DK) (FX Grid to Provider)" on page 41<br>■ Reworked field order and repeated groups in "Market Data Request (FX Grid to Provider)" on page 35.<br>■ Removed `MaturityDate` (#541) from "Market Data Request (FX Grid to Provider)".<br>■ Updated description of `FutSettDate` (#64) in "New Order – Single (FX Grid to Provider)" on page 38. |

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

| Date | API Version | Enhancements |
|---|---|---|
| July 2011 (continued) | 1.6 | ■ The Party component block has been added to the to capture the ClientTag in the `PartyID` (#448) field of the "New Order – Single (FX Grid to Provider)" on page 38.<br>■ To support customer requests for full or partial fills, added `MinQty` (#110) to "New Order – Single (FX Grid to Provider)" on page 38.<br>■ Added new values (GTD, FOK) to `TimeInForce` (#59) in "New Order – Single (FX Grid to Provider)" on page 38<br>■ Added "Partial Fills" on page 31.<br>■ Expanded the following sections:<br>□ "Supported Order Types" on page 30 (added limit and market orders)<br>□ "Order Expiry" on page 30 added GTD and FOK)<br>■ Added the following fields to "New Order – Single (FX Grid to Provider)" on page 38:<br>□ `BidPx` (#132)<br>□ `OfferPx` (#133)<br>□ `BidSpotRate` (#188)<br>□ `BidForwardPoints` (#189)<br>□ `OfferSpotRate` (#190)<br>□ `OfferForwardPoints` (#191) |
| January 2011 (3.8v3) | 1.5 | ■ Definition of `ExpireTime` (#126) changed to clarify that the value is the relative expiry time. See "Trading Messages" on page 33.<br>■ Required status of `TradeDate` (#75) field changed to not required in "Execution Report (Provider to FX Grid)" on page 42.<br>■ Descriptions of the following fields changed to clarify how to determine the price of a fill:<br>□ "LastPx" on page 44<br>□ "AvgPx" on page 44 |
| October 2010 (3.6.1v1) | 1.0 | ■ Added `HandlInst` (#21) to New Order – Single message. See "New Order – Single (FX Grid to Provider)" on page 38.<br>■ Added `LastQty` (#32) to Execution Report message. For a fill or partial fill, this field must be specified and must not be set to zero or the message is rejected. See "LastQty" on page 43.<br>■ Clarified the value of in `TradingSessionID` (#336) in Trading Session Status. Can be any string. See "TradingSessionID" on page 34.<br>■ Added information about canceling ESP prices. See "Quote Cancel" on page 27. |
| August 2010 | 1.0 | ■ Added clarification of behavior when a value date is not provided for `MaturityDate` (#541) on Market Data Snapshot/Full Refresh messages and `FutSettDate` (#64) Execution Report messages for ESP spot trades. See "Market Data Snapshot/Full Refresh (Provider to FX Grid)" on page 36 and "Execution Report (Provider to FX Grid)" on page 42. |
| December 2009 | 1.0 | ■ First version<br>■ Added requirement that the `MDReqID` (#262) field value should not contain the ampersand character "@". See "Market Data Request (FX Grid to Provider)" on page 35. |

APPENDIX B

# Examples

# B.1 Session Management

## B.1.1 Logon (Order Session)

```
8=FIX.4.3|9=78|35=A|34=1|49=demo.fxgrid|52=20140624-
09:50:57.031|56=order.ProviderOrg|98=0|108=30|141=N|10=107|
```

## B.1.2 Logon (Price Session)

```
8=FIX.4.3|9=78|35=A|34=1|49=demo.fxgrid|52=20140624-
09:50:57.032|56=price.ProviderOrg|98=0|108=30|141=Y|10=090|
```

## B.1.3 Logon (Response, Order Session)

```
8=FIX.4.3|9=78|35=A|34=1|49=order.ProviderOrg|52=20140624-
09:50:56.967|56=demo.fxgrid|98=0|108=30|141=Y|10=124|
```

### B.1.4 Logon (Response, Price Session)

```
8=FIX.4.3|9=78|35=A|34=1|49=price.ProviderOrg|52=20140624-
09:50:56.968|56=demo.fxgrid|98=0|108=30|141=Y|10=107|
```

### B.1.5 Logout (Order Session)

```
8=FIX.4.3|9=78|35=5|34=109|49=demo.fxgrid|52=20140624-
13:58:16.225|56=order.ProviderOrg|10=133|
```

### B.1.6 Logout (Price Session)

```
8=FIX.4.3|9=77|35=5|34=93|49=demo.fxgrid|52=20140624-
13:58:16.225|56=price.ProviderOrg|10=087|
```

### B.1.7 Logout (Response, Failed Logon)

```
8=FIX.4.4|9=75|35=5|34=2889|49=demo.provider|52=20140704-
21:01:00.971|56=price.ProviderOrg|58=Unknown user ID|10=100|
```

### B.1.8 Logout (Response, Order Session)

```
8=FIX.4.3|9=78|35=5|34=135|49=order.ProviderOrg|52=20140624-
13:58:16.155|56=demo.fxgrid|10=134|
```

### B.1.9 Logout (Response, Price Session)

```
8=FIX.4.3|9=80|35=5|34=17098|49=price.ProviderOrg|52=20140624-
13:58:16.155|56=demo.fxgrid|10=240|
```

### B.1.10 Heartbeat (Incoming)

```
8=FIX.4.3|9=66|35=0|34=4|49=order.ProviderOrg|52=20140624-
09:51:27.963|56=demo.fxgrid|10=077|
```

### B.1.11 Heartbeat (Outgoing)

```
8=FIX.4.3|9=66|35=0|34=5|49=demo.fxgrid|52=20140624-
09:52:27.989|56=order.ProviderOrg|10=087|
```

### B.1.12 Resend Request

```
8=FIX.4.3|9=107|35=2|34=240|49=price.ProviderOrg|52=20140704-
10:01:56.324|56=demo.fxgrid|7=2647|16=0|10=229|
```

### B.1.13 Sequence Reset

```
8=FIX.4.3|9=137|35=4|34=2672|43=Y|49=demo.fxgrid|52=20140704-
10:02:04.098|56=price.ProviderOrg|122=20140704-10:02:04|36=2680|123=Y|10=214|
```

### B.1.14 Session Level Reject

```
8=FIX.4.3|9=169|35=3|34=11|49=order.ProviderOrg|50=FI220LE|52=20140705-
14:14:34.070|56=demo.fxgrid|57=DBNB|128=orgID|45=13|58=Invalid tag
number|371=50|372=8|373=0|10=113|
```

### B.1.15 Test Request

```
8=FIX.4.3|9=105|35=1|34=1089|49=demo.fxgrid|52=20140705-
09:06:30.882|56=price.ProviderOrg|112=TEST|10=034|
```

## B.2 Market Data

### B.2.1 Market Data Request

```
8=FIX.4.3|9=159|35=V|34=5|49=demo.fxgrid|52=20140518-
21:51:48.272|56=price.ProviderOrg|262=FXLP71300517de30117|263=1|264=0|265=0|75
40=Stream1|146=1|55=EUR/USD|460=4|267=2|269=0|269=1|10=183|
```

### B.2.2 Market Data Request Reject

```
8=FIX.4.3|9=132|35=Y|34=579967|49=price.ProviderOrg|52=20140629-
09:07:10.231|56=demo.fxgrid|58=USD/NOK is not
supported|262=FXLP130d828bd0c155|10=041|
```

### B.2.3 Market Data Incremental Refresh

```
8=FIX4.3|9=1962|35=X|34=459648|49=price.ProviderOrg|52=20140705-
06:59:59.640|56=demo.fxgrid|268=29|279=0|269=0|55=GBP/USD|461=RCSXXX|270=1.469
34|271=20000000|63=0|279=0|269=1|55=GBP/USD|461=RCSXXX|270=1.47084|271=2000000
0|63=0|279=0|269=0|55=GBP/USD|461=RCSXXX|270=1.46929|271=25000000|63=0|279=2|2
69=1|55=GBP/USD|461=RCSXXX|270=1.47034|271=10000000|63=0|279=2|269=1|55=GBP/US
D|461=RCSXXX|270=1.47034|271=10000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|2
70=1.47039|271=15000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47044|271
=20000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47044|271=12000000|63=0
|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47044|271=12000000|63=0|279=2|269=1|5
5=GBP/USD|461=RCSXXX|270=1.47049|271=17000000|63=0|279=2|269=1|55=GBP/USD|461=
RCSXXX|270=1.47054|271=22000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.4
7054|271=14000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47054|271=14000
000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47059|271=19000000|63=0|279=2
|269=1|55=GBP/USD|461=RCSXXX|270=1.47064|271=24000000|63=0|279=2|269=1|55=GBP/
USD|461=RCSXXX|270=1.47064|271=16000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX
|270=1.47064|271=16000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47069|2
71=21000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47074|271=26000000|63
=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47074|271=18000000|63=0|279=2|269=1
|55=GBP/USD|461=RCSXXX|270=1.47074|271=18000000|63=0|279=2|269=1|55=GBP/USD|46
1=RCSXXX|270=1.47079|271=23000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1
.47084|271=28000000|63=0|279=2|269=1|55=GBP/USD|461=RCSXXX|270=1.47084|271=200
00000|63=0|279=0|269=1|55=GBP/USD|461=RCSXXX|270=1.47089|271=25000000|63=0|279
=0|269=0|55=GBP/USD|461=RCSXXX|270=1.46924|271=30000000|63=0|279=0|269=1|55=GB
P/USD|461=RCSXXX|270=1.47094|271=30000000|63=0|279=0|269=0|55=GBP/USD|461=RCSX
XX|270=1.46934|271=20000000|63=0|279=0|269=1|55=GBP/USD|461=RCSXXX|270=1.47084
|271=20000000|63=0|10=027|
```

### B.2.4 Market Data Snapshot/Full Refresh (Multi-price Quote)

```
8=FIX.4.3|9=505|35=W|34=4|49=price.ProviderOrg|52=20140518-
21:51:53.884|56=demo.fxgrid|55=EUR/USD|262=FXLP71300517de30117|268=6|269=0|270
=1.403|15=EUR|271=1000000|276=A|282=1|299=26
903010307270408449|290=0|269=0|270=1.4046|15=EUR|271=5000000|276=A|282=1|299=2
7 221090307220403130|290=0|269=0|270=0|15=EUR|271=0|276=A|282=1|299=28
4|290=0|269=1|270=1.4074|15=EUR|271=1000000|276=A|282=1|299=26
783080909280801987|290=0|269=1|270=0|15=EUR|271=0|276=A|282=1|299=27
3|290=0|269=1|270=0|15=EUR|271=0|276=A|282=1|299=28 5|290=0|460=4|10=045|
```

### B.2.5 Market Data Snapshot/Full Refresh (Multi-price Quote, Example 2)

```
8=FIX.4.3|9=505|35=W|34=4|49=price.ProviderOrg|52=20140518-
21:51:53.884|56=demo.fxgrid|55=EUR/USD|262=FXLP71300517de30117|268=6|269=0|270
=1.403|15=EUR|271=1000000|276=A|282=1|299=26
903010307270408449|290=0|269=0|270=1.4046|15=EUR|271=5000000|276=A|282=1|299=2
7 221090307220403130|290=0|269=0|270=0|15=EUR|271=0|276=A|282=1|299=28
4|290=0|269=1|270=1.4074|15=EUR|271=1000000|276=A|282=1|299=26
783080909280801987|290=0|269=1|270=0|15=EUR|271=0|276=A|282=1|299=27
3|290=0|269=1|270=0|15=EUR|271=0|276=A|282=1|299=28 5|290=0|460=4|10=045|
```

### B.2.6 Market Data Snapshot/Full Refresh (Multi-tier Quote)

```
8=FIX.4.3|9=587|35=W|34=211847|49=price.ProviderOrg|52=20140530-
15:00:00.511|56=demo.fxgrid|55=USD/JPY|262=PROVIDER1303fbc1e6211|460=4|541=201
40601|268=6|269=0|270=80.919|15=USD|271=1000000|276=A|282=PROVIDER|299=1516753
97639_
B|290=1|269=1|270=80.928|15=USD|271=1000000|276=A|282=PROVIDER|299=15167539763
9_
O|290=1|269=0|270=80.92|15=USD|271=3000000|276=A|282=PROVIDER|299=15167539763
9_
B1|290=2|269=0|270=80.91|15=USD|271=5000000|276=A|282=PROVIDER|299=15167539763
9_
B2|290=3|269=1|270=80.93|15=USD|271=3000000|276=A|282=PROVIDER|299=15167539763
9_
O1|290=2|269=1|270=80.94|15=USD|271=5000000|276=A|282=PROVIDER|299=15167539763
9_O2|290=3|10=161|
```

## B.3 Trading

### B.3.1 Order Reject (New Order – Single, Rejected Order)

```
8=FIX.4.3|9=219|35=D|34=5| 49=demo.fxgrid|52=20140518-
21:28:47.037|56=order.ProviderOrg|1=ProviderAccnt|11=FXI57748726|15=EUR|21=1|3
8=1000000|40=D|44=1.4048|54=2|55=EUR/USD|59=3|60=20140518-21:28:47.036|117=27
718040708200504334|167=FOR|7540=Stream1|10=225|
```

### B.3.2 Order Reject (Reject, Execution Report)

```
8=FIX.4.3|9=253|35=8|34=6|49=order.ProviderOrg|52=20140518-
21:28:47.081|56=demo.fxgrid|1=ProviderAccnt|6=0|11=FXI57748726|14=0|17=1013057
5412707914|32=0|37=CA-FXI57748726|38=1000000|39=8|54=2|55=EUR/USD|58=Rate
```

```
Moved|60=20140518
21:28:47|75=19700101|119=0|120=EUR|150=8|151=0|167=FOR|10=090|
```

## B.3.3 Order Timeout (Execution Report)

```
8=FIX.4.3|9=92|35=OT|34=412|49=demo.fxgrid|52=20140601-
00:54:25.130|56=order.ProviderOrg|11=FXI107111|10=157|
```

## B.3.4 Trade, Full Fill, Multi Fill, New Order – Single

```
8=FIX.4.3|9=219|35=D|34=4|49=demo.fxgrid|52=20140517-
22:17:16.035|56=order.ProviderOrg|1=ProviderAccnt|11=FXI57677008|15=EUR|21=1|3
8=5000000|40=D|44=1.4047|54=2|55=EUR/USD|59=3|60=20140517-22:17:16.034|117=27
299010107270301260|167=FOR|7540=Stream1|10=196|
```

## B.3.5 Trade, Full Fill, Multi Fill, ExecutionReport, Trade 1

```
8=FIX.4.3|9=302|35=8|34=4|49=order.ProviderOrg|52=20140517-
22:17:16.085|56=demo.fxgrid|1=ProviderAccnt|6=1.4047|11=FXI57677008|14=1000000
|17=FXI57677008-1305670636084-1|31=1.4047|32=1000000|37=FXI57677008AC-
|38=5000000|39=1|54=2|55=EUR/USD|60=20140517-
22:17:16|75=20140518|119=1000000|120=EUR|150=1|151=4000000|167=FOR|10=211|
```

## B.3.6 Trade, Full Fill, Multi Fill, ExecutionReport, Trade 2

```
8=FIX.4.3|9=296|35=8|34=5|49=order.ProviderOrg|52=20140517-
22:17:16.088|56=demo.fxgrid|1=ProviderAccnt|6=1.4047|11=FXI57677008|14=5000000
|17=FXI57677008-1305670636087-2|31=1.4047|32=4000000|37=FXI57677008AC-
|38=5000000|39=2|54=2|55=EUR/USD|60=20140517-
22:17:16|75=20140518|119=4000000|120=EUR|150=F|151=0|167=FOR|10=207|
```

## B.3.7 Trade, Full Fill, Single Fill, New Order – Single

```
8=FIX.4.3|9=221|35=D|34=54|49=demo.fxgrid|52=20140505-
21:35:35.375|56=order.ProviderOrg|1=ProviderAccnt1|11=FXI57002344|15=EUR|21=1|
38=1000000|40=D|44=1.45455|54=2|55=EUR/USD|59=3|60=20140505-
21:35:35.375|64=20140510|117=FXLP1312fc20dd75936|167=FOR|7540=Stream1|10=056|
```

## B.3.8 Trade, Full Fill, Single Fill, Execution Report, Pending New

```
8=FIX.4.3|9=229|35=8|34=65|49=order.ProviderOrg|52=20140505-
21:35:34.528|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002344|14=0|17=FXI570
02344|37=FXI57002344|38=1000000|39=A|44=1.45455|54=2|55=EUR/USD|60=20140505-
21:35:34|119=0|120=USD|150=0|151=0|167=FOR|10=180|
```

## B.3.9 Trade, Full Fill, Single Fill, Execution Report, New

```
8=FIX.4.3|9=229|35=8|34=66|49=order.ProviderOrg|52=20140505-
21:35:34.530|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002344|14=0|17=FXI570
02344|37=FXI57002344|38=1000000|39=0|44=1.45455|54=2|55=EUR/USD|60=20140505-
21:35:34|119=0|120=USD|150=0|151=0|167=FOR|10=157|
```

## B.3.10 Trade, Full Fill, Single Fill, Execution Report, Trade

```
8=FIX.4.3|9=251|35=8|34=67|49=order.ProviderOrg|52=20140505-
21:35:34.986|56=demo.fxgrid|6=1.45455|11=FXI57002344|14=1000000|17=S000000A057
7ECA9E|31=1.45455|32=1000000|37=FXI57002344|38=1000000|39=2|54=2|55=EUR/USD|60
=20140505-16:35:34|119=1454550|120=USD|150=F|151=0|167=FOR|10=049|
```

## B.3.11 Trade, Partial Fill, Multi Fill, New Order – Single

```
8=FIX.4.3|9=221|35=D|34=51|49=demo.fxgrid|52=20140505-
21:34:17.255|56=order.ProviderOrg|1=ProviderAccnt1|11=FXI57002340|15=EUR|21=1|
38=1000000|40=D|44=1.45453|54=2|55=EUR/USD|59=3|60=20140505-
21:34:17.255|64=20140510|117=FXLP1312fc20dd75936|167=FOR|7540=Stream1|10=039|
```

## B.3.12 Trade, Partial Fill, Multi Fill, Execution Report, Order, Pending New

```
8=FIX.4.3|9=229|35=8|34=59|49=order.ProviderOrg|52=20140505-
21:34:16.413|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002340|14=0|17=FXI570
02340|37=FXI57002340|38=1000000|39=A|44=1.45453|54=2|55=EUR/USD|60=20140505-
21:34:16|119=0|120=USD|150=0|151=0|167=FOR|10=160|
```

## B.3.13 Trade, Partial Fill, Multi Fill, Execution Report, Order, New

```
8=FIX.4.3|9=229|35=8|34=60|49=order.ProviderOrg|52=20140505-
21:34:16.416|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002340|14=0|17=FXI570
02340|37=FXI57002340|38=1000000|39=0|44=1.45453|54=2|55=EUR/USD|60=20140505-
21:34:16|119=0|120=USD|150=0|151=0|167=FOR|10=138|
```

## B.3.14 Trade, Partial Fill, Multi Fill, Execution Report, Trade 1

```
8=FIX.4.3|9=256|35=8|34=61|49=order.ProviderOrg|52=20140505-
21:34:16.877|56=demo.fxgrid|6=1.45453|11=FXI57002340|14=125000|17=S000000A052D
6F5AE|31=1.45453|32=125000|37=FXI57002340|38=1000000|39=1|54=2|55=EUR/USD|60=2
0140505-16:34:16|119=181816.25|120=USD|150=1|151=875000|167=FOR|10=027|
```

## B.3.15 Trade, Partial Fill, Multi Fill, Execution Report, Trade 2

```
8=FIX.4.3|9=256|35=8|34=62|49=order.ProviderOrg|52=20140505-
21:34:18.568|56=demo.fxgrid|6=1.45453|11=FXI57002340|14=250000|17=S000000A052F
0BDAD|31=1.45453|32=125000|37=FXI57002340|38=1000000|39=1|54=2|55=EUR/USD|60=2
0140505-16:34:18|119=181816.25|120=USD|150=1|151=750000|167=FOR|10=026|
```

## B.3.16 Trade, Partial Fill, Multi Fill, Execution Report, Order Expiry

```
8=FIX.4.3|9=247|35=8|34=63|49=order.ProviderOrg|52=20140505-
21:34:36.098|56=demo.fxgrid|1=ProviderAccnt1|6=1.45453|11=FXI57002340|14=25000
0|17=FXI57002340|37=FXI57002340|38=1000000|39=C|44=1.45453|54=2|55=EUR/USD|60=
20140505-21:34:16|119=363632.5|120=USD|150=C|151=0|167=FOR|10=080|
```

## B.3.17 Trade, Partial Fill, Single Fill, New Order – Single

```
8=FIX.4.3|9=221|35=D|34=43|49=demo.fxgrid|52=20140505-
21:31:01.762|56=order.ProviderOrg|1=ProviderAccnt1|11=FXI57002334|15=EUR|21=1|
```

```
38=1000000|40=D|44=1.45444|54=2|55=EUR/USD|59=3|60=20140505-
21:31:01.762|64=20140510|117=FXLP1312fc20dd75936|167=FOR|7540=Stream1|10=029|
```

## B.3.18 Trade, Partial Fill, Single Fill, Execution Report, Order, Pending New

```
8=FIX.4.3|9=229|35=8|34=48|49=order.ProviderOrg|52=20140505-
21:31:00.921|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002334|14=0|17=FXI570
02334|37=FXI57002334|38=1000000|39=A|44=1.45444|54=2|55=EUR/USD|60=20140505-
21:31:00|119=0|120=USD|150=0|151=0|167=FOR|10=151|
```

## B.3.19 Trade, Partial Fill, Single Fill, Execution Report, Order, New

```
8=FIX.4.3|9=229|35=8|34=49|49=order.ProviderOrg|52=20140505-
21:31:00.923|56=demo.fxgrid|1=ProviderAccnt1|6=0|11=FXI57002334|14=0|17=FXI570
02334|37=FXI57002334|38=1000000|39=0|44=1.45444|54=2|55=EUR/USD|60=20140505-
21:31:00|119=0|120=USD|150=0|151=0|167=FOR|10=137|
```

## B.3.20 Trade, Partial Fill, Single Fill, Execution Report, Trade

```
8=FIX.4.3|9=253|35=8|34=50|49=order.ProviderOrg|52=20140505-
21:31:01.382|56=demo.fxgrid|6=1.45444|11=FXI57002334|14=125000|17=S000000A0472
FF846|31=1.45444|32=125000|37=FXI57002334|38=1000000|39=1|54=2|55=EUR/USD|60=2
0140505-16:31:01|119=181805|120=USD|150=1|151=875000|167=FOR|10=083|
```

## B.3.21 Trade, Partial Fill, Single Fill, Execution Report, Order Expiry

```
8=FIX.4.3|9=245|35=8|34=51|49=order.ProviderOrg|52=20140505-
21:31:07.263|56=demo.fxgrid|1=ProviderAccnt1|6=1.45444|11=FXI57002334|14=12500
0|17=FXI57002334|37=FXI57002334|38=1000000|39=C|44=1.45444|54=2|55=EUR/USD|60=
20140505-21:31:00|119=181805|120=USD|150=C|151=0|167=FOR|10=221|
```

## B.3.22 Trade, Term Currency, New Order – Single

```
8=FIX.4.3|9=219|35=D|34=4|49=demo.fxgrid|52=20140518-
21:52:29.916|56=order.ProviderOrg|1=ProviderAccnt|11=FXI57748770|15=USD|21=1|3
8=5000000|40=D|44=1.4045|54=1|55=EUR/USD|59=3|60=20140518-
21:52:29.916|117=27800090007270703184|167=FOR|7540=Stream1|10=230|
```

## B.3.23 Trade, Term Currency, Execution Report, Trade 1

```
8=FIX.4.3|9=303|35=8|34=4|49=order.ProviderOrg|52=20140518-
21:52:29.962|56=demo.fxgrid|1=ProviderAccnt|6=1.4045|11=FXI57748770|14=1404500
|17=FXI57748770-1305755549961-55|31=1.4045|32=1404500|37=FXI57748770AC-
|38=5000000|39=1|54=1|55=EUR/USD|60=20140518-
21:52:29|75=20140519|119=1404500|120=USD|150=1|151=3595500|167=FOR|10=110|
```

## B.3.24 Trade, Term Currency, Execution Report, Trade 2

```
8=FIX.4.3|9=303|35=8|34=5|49=order.ProviderOrg|52=20140518-
21:52:29.962|56=demo.fxgrid|1=ProviderAccnt|6=1.4045|11=FXI57748770|14=2809000
|17=FXI57748770-1305755549961-56|31=1.4045|32=1404500|37=FXI57748770AC-
|38=5000000|39=1|54=1|55=EUR/USD|60=20140518-
21:52:29|75=20140519|119=1404500|120=USD|150=1|151=2191000|167=FOR|10=103|
```

### B.3.25 Trade, Term Currency, Execution Report, Order Expiry

```
8=FIX.4.3|9=260|35=8|34=6|49=order.ProviderOrg|52=20140518-
21:52:29.963|56=demo.fxgrid|1=ProviderAccnt|6=1.4045|11=FXI57748770|14=2809000
|17=COMPLETE|32=0|37=COMPLETEFXI57748770|38=5000000|39=C|54=1|55=EUR/USD|60=20
140518-21:52:29|75=19700101|119=0|120=USD|150=C|151=0|167=FOR|10=057|
```

## B.4 Trading Session

### B.4.1 Trading Session Status Request (Order Session)

```
8=FIX.4.3|9=95|35=g|34=2|49=demo.fxgrid|52=20140624-
09:50:57.032|56=order.ProviderOrg|263=0|335=Order1308909057032|10=246|
```

### B.4.2 Trading Session Status Request (Price Session)

```
8=FIX.4.3|9=95|35=g|34=2|49=demo.fxgrid|52=20140624-
09:50:57.032|56=price.ProviderOrg|263=0|335=Price1308909057032|10=219|
```

### B.4.3 Trading Session Status (Open, Order Session)

```
8=FIX.4.3|9=111|35=h|34=2|49=order.ProviderOrg|52=20140505-
21:05:07.185|56=demo.fxgrid|325=N|335=Order1304629507988|336=20140505|340=2|10
=096|
```

### B.4.4 Trading Session Status (Open, Price Session)

```
8=FIX.4.3|9=111|35=h|34=2|49=price.ProviderOrg|52=20140505-
21:05:07.189|56=demo.fxgrid|325=N|335=Price1304629507988|336=20140505|340=2|10
=073|
```