B2C2

FIX SPECIFICATION

Developer Guide

Version 2.2.3

05/03/2019

Version History

Version	Date	Comments
1.0.0	21/11/2017	Initial Version.
2.0.0	01/02/2018	Added support for Fill or Kill orders.
2.1.0	10/09/2018	Added LastPx and LastQty to Execution Reports Minor errata.
2.2.0	28/11/2018	Added Drop Copy section. Added Account tag to order flow. Added TransactTime to Quote message. Added Security List Request and Security List messages to trading session. Minor other clean up and clarifications, especially around tag use in cases of rejections and unsubscriptions.
2.2.1	10/01/2019	Errata on EncryptMethod valid value N → 0
2.2.2	21/01/2019	Tag clarifications in Drop Copy section. Addition of Text tag to MarketDataRequestReject.
2.2.3	05/03/2019	Added clarification over empty Market Data – Snapshot / Full Refresh messages

Table of Contents

This document contains the FIX interface specification for B2C2's trading platform. The document is intended to be used by clients as a technical reference when building systems that interact with B2C2 through FIX.

Using B2C2 FIX API you can:

1. Subscribe to streaming prices in multiple quantity levels *
2. Issue requests for quotes (RFQs)
3. Trade on an RFQ
4. Send fill or kill (FOK) limit orders
5. Receive trade executions on a read only feed

* Prices are indicative, you still need to request a quote before trading

The B2C2 FIX API supports a modified FIX Protocol 4.4 (herein known as FIX 4.4-B2C2), the modifications are optional and are fully backwards compatible with the original 4.4 protocol. The modification only impacts subscribing to market data.

For the full details of the amendment to the FIX protocol please see the custom fields and messages section.

1.1 Session Types

The B2C2 FIX api supports three session types:

- **Trading Session** - For submitting quote requests and orders
- **Market Data Session** - For subscribing to streaming prices
- **Drop Copy Session** – For passively receiving trade executions, only

Both the Trading and Drop Copy session types support Security List messaging.

1.2 Connectivity

The B2C2 FIX electronic pricing and execution platform is hosted in AWS. Access to the platform is supported by the following connectivity options:

- **TLS/SSL encrypted TCP connection over Internet** – strongly preferred, very quick to set up
- TCP/IP connection over Internet – Sandbox / UAT only, very quick to set up
- AWS VPC Endpoints for selected regions – on request
- Public data centre cross connect – planned, but not available currently

Server	Session	Host	Port	Protocol
Sandbox / UAT	Trading	Available on Request	Available on Request	FIX 4.4
Sandbox/ UAT	Market Data	Available on Request	Available on Request	FIX 4.4-B2C2
Sandbox/ UAT	Drop Copy	Available on Request	Available on Request	FIX 4.4
Production / Live	Trading	Available on Request	Available on Request	FIX 4.4
Production / Live	Market Data	Available on Request	Available on Request	FIX 4.4-B2C2
Production / Live	Drop Copy	Available on Request	Available on Request	FIX 4.4

1.3 Certification Process

All clients will have to go through conformance tests to ensure that the client's implementation conforms to the specifications outlined in this document. The certification process will test all aspects of subscribing to market data, requesting quotes and executing quotes.

Once the conformance tests have been passed, the client may be permitted to start trading on the production engine.

1.4 Operating Hours

The B2C2 operating hours are described in the following schedule, all times are in UTC

	Time Zone	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Open	UTC	00:00	00:00	00:00	00:00	00:00	00:00	00:00
Close	UTC	24:00	24:00	24:00	24:00	24:00	24:00	24:00

1.5 FIX Session Schedule

B2C2 does not reset sequence numbers on any schedule. It is left to the client to reset sequence numbers on Logon using ResetSeqNumFlag(141) = Y, per the following recommendations:

- If a connection to a trading is disconnected the client should login with the next sequence number.
- If a connection is lost to a market data session the client should login using a sequence number of 1. Once successfully logged in the client should send a market data request for any price stream that it is interested in.
- If a connection is lost to a drop copy session the client should login using a sequence number of 1. Once successfully logged in the client will receive all trades executed from the start of that Business Day followed by any new executions in near real time.

1.6 Quoting & Trading Conventions

In all application messages, the symbol field defines the currency pair and instrument type.

The symbol format is (BASECCY)(COUNTERCCY).(TYPE) where:

- (BASECCY) is the base currency
- (COUNTERCCY) is the counter currency
- (TYPE) can take the values SPOT or CFD to indicate the instrument type.

E.g the symbol BTCUSD.SPOT is for the spot market Bitcoin against the US dollar.

All of the quantity fields will be in terms of the base currency.

Rates are expressed as units of counter currency per unit of base currency. E.g an order to buy 10 BTCUSD.SPOT @ 9800 means buying 10 BTC with 98,000 USD.

It is important to note that when trading crypto vs fiat that depending on the traded quantity it can lead to sub-penny increments in fiat currency proceeds, these proceeds are not rounded.

E.g if you buy 0.0123 BTCUSD @ $10 then you will pay $0.123 (note that it isn’t rounded).

1.7 FIX Tag Order

FIX header tags should be sent in order defined in this specification.

1.8 Business Day

The term Business Day as used in this document starts and ends at 5pm Eastern Daylight Time (EDT): local time on the US East Coast.

1.9 Message Flow

1.9.1 Supported Messages

The following administrative messages will be supported:

- Logon
- Heartbeat
- Test Request
- Resend Requesting
- Sequence Reset
- Session-level Reject
- Logout

The following application level messages will be supported in the market data session:

- Market Data Request
- Market Data Request Reject
- Market Data – Snapshot / Full Refresh
- Security List Request
- Security List

The following application messages will be supported in the trading session:

- Quote Request
- Quote Request Reject
- Quote
- New Order Single
- Execution Report
- Security List RequestSecurity List

The following application messages will be supported in the drop copy session:

- Execution Report

2 Messages

2.1 Session Messages

2.1.1 Standard Header

Every message, whether administrative or application, is preceded by a StandardHeader. The header identifies the message type, length, destination, sequence number, origination point and time.

Tag	Name	Req	Description
8	BeginString	Y	Identifies the beginning of a new message and protocol version. Valid value: FIX.4.4
9	BodyLength	Y	Message of message body not including the header/trailer
35	MsgType	Y	Defines the message type.
34	MsgSeqNum	Y	Integer message sequence number.
49	SenderCompID	Y	Used to identify the firm sending the message and will be agreed upon during the onboarding process.
43	PossDupFlag	N	Required if this message is a retransmission of a message with the same MsgSeqNum. Valid values: • Y = Possible duplicate, • N = Original transmission
52	SendingTime	Y	UTC Timestamp of when the current message was transmitted
56	TargetCompId	Y	Used to identify the firm receiving the message. Valid value: Dependant on environment and session types
122	OrigSendingTime	N	Required for messages resent as a result of a ResendRequest. It is the SendingTime of the message when it was initially sent

2.1.2 Standard Trailer

The StandardTrailer is required on on every message.

Tag	Name	Req	Description

10	CheckSum	Y	Three byte, simple checksum. Calculated for the entire message (i.e. includes all header fields and body)

2.1.3 Heartbeat <0>

The Heartbeat message monitors the status of the communication link and identifies when the last of a string of messages was not received.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 0
112	TestReqID	N	Required when responding to a Test Request message.
	<MessageTrailer>	Y	

2.1.4 Logon <A>

The Logon message is sent by the client to initiate a session. The Logon message must be the first message sent by a remote counterparty initiating a FIX session. A message of the same type will be sent in response to acknowledge the logon.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = A
98	EncryptMethod	Y	Valid value: 0 = None
108	HeartBtInt	Y	Heartbeat interval in seconds
141	ResetSeqNumFlag	N	Indicates both sides should reset sequence numbers
553	Username	Y	Your API token, retrieved from the B2C2 website
	<MessageTrailer>	Y	

2.1.5 Test Request <1>

The TestRequest message forces a heartbeat from the opposing application in order to verify the communication link status.

On receipt of a TestRequest the recipient should reply with a Heartbeat message containing the corresponding TestReqID.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 1

112	TestReqID	Y	
	<MessageTrailer>	Y	

2.1.6 Resend Request <2>

The ResendRequest message should be sent to initiate the retransmission of messages if a sequence number gap is detected due to lost message or as a function of the initialization process.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 2
7	BeginSeqNo	Y	Message sequence number of first message in range to be resent
16	EndSeqNo	Y	Message sequence number of last message in range to be resent
	<MessageTrailer>	Y	

2.1.7 Reject <3>

The Reject message will be used to reject any messages received that deviate from this specification.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 3
45	RefSeqNum	Y	MsgSeqNum <34> of rejected message
	<MessageTrailer>	Y	

2.1.8 Sequence Reset <4>

The SequenceReset message can be sent from both sides to reset the sequence number

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 4
123	GapFillFlag	N	
36	NewSeqNo	Y	

	<MessageTrailer>	Y	

2.1.9 Logout <5>

The Logout message is sent by either side to initiate a session termination. A logout message should replied to with the same message type to confirm the session termination. Failure to do so before terminating the session is an error.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 5
58	Text	N	The reason for logging out
	<MessageTrailer>	Y	

2.2 Securities Information - Requests and Responses

2.2.1 Security List Request <x>

The Security List Request message is used to return a list of securities that the client has permission to trade contained within a Security List message.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = x
320	SecurityReqID	Y	
559	SecurityListRequestType	Y	Valid value: 0 = Symbol <55>
	<MessageTrailer>	Y	

2.2.2 Security List <y>

The Security List message is sent in response to a Security List Request and contains all the securities that the client is permitted to trade.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = y
320	SecurityReqID	Y	
322	SecurityResponseId	Y	Unique identifier for the Security List <y> response
560	SecurityRequestResult	Y	Result of the security list request identified by SecurityReqId <320> Valid values: • 0 = Valid Request, • 1 = Invalid Request
393	TotNoRelatedSym	Y	Indicates the total number of securities to be returned for this request
146	NoRelatedSym	Y	Indicates the number of securities to be returned in this response
=>55	Symbol	Y	The symbol of the traded security
	<MessageTrailer>	Y	

2.3 Market Data - Requests and Responses

2.3.1 Market Data Request <V>

The Market Data Request message subscribes the current session to a stream of Market Data - Snapshot/Full Refresh messages. The NoQuantityLevels field is a custom field not in the original FIX 4.4 protocol, however it is optional ensuring backwards compatibility with FIX 4.4 It allows to subscribe to various levels of quantity, if no value is specified then the Market Data Snapshot will subscribe to the quantity that currently corresponds to a notional value of 1000 USD and 5000 USD. For the full details of the modification please see the custom fields and messages section.

Note: for unsubscriptions (263=2), the only additional tag allowed is MdReqID.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = V
262	MDReqID	Y	Unique identifier for the market data request
263	SubscriptionRequestType	Y	Indicates what type of response is expected. Valid values: • 1 = Snapshot + Updates (Subscribe), • 2 = Unsubscribe
264	MarketDepth	Y	Depth of book to receive snapshot and updates for. Valid value: 0 = Full Book
265	MDUpdateType	Y	Specifies the type of market data update. Valid Value: 0 = Full Refresh
267	NoMDEntryTypes	Y	The number of MDEntryType fields requested
=> 269	MDEntryType	Y	The type of market data entry to receive snapshots and updates for. Value values: 0 = Bid, 1 = Offer
146	NoRelatedSym	Y	The number of symbols requested
=> 55	Symbol	Y	The symbol to get data for
=> 8000	NoQuantityLevels	N	The number of quantity levels to subscribe to

			prices for, the maximum number of levels is 2
==> 53	Quantity	C	• Maximum precision: 4 decimals • Minimum: 0.1 • Maximum: by client
	<MessageTrailer>	Y	

2.3.2 Market Data Request Reject <Y>

The Market Data Request Reject message is used to reject a request for market data due to either an invalid request or technical issues.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = Y
262	MDReqID	Y	Refers to the MDReqID of the request being rejected.
281	MDReqRejReason	N	The reason why the request was rejected.
58	Text	N	Description of the rejection.
	<MessageTrailer>	Y	

2.3.3 Market Data - Snapshot/Full Refresh <W>

The Market Data - Snapshot/Full Refresh message is used to transmit updates to the indicative prices / order book.

Note: NoMDEntries might be absent or set to zero to signify an upstream pricing issue, i.e. the most recent refresh where prices and quantities were sent is no longer valid.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = W
262	MDReqID	Y	Refers to the MDReqID of the initial request.
55	Symbol	Y	Symbol of the market data entry
268	NoMDEntries	N	The number of entries following
=> 269	MDEntryType	Y	The type of market data update.

			Valid values: - 0 = Bid, - 1 = Offer
=> 270	MDEntryPx	Y	Price of the market data entry
=> 271	MDEntrySize	Y	Quantity of the market data entry
	<MessageTrailer>	Y	

2.4 Request for Quote (RFQ) - Requests and Responses

2.4.1 Quote Request <R>

The Quote Request message is used to request a quote (RFQ) that is executable. Requesting a quote invalidates all previous RFQ's that have yet to be executed.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = R
131	QuoteReqID	Y	The unique identifier for the request for quote (RFQ)
146	NoRelatedSym	Y	The number of symbols to request a quote for. Valid value: 1
=> 55	Symbol	Y	The symbol to request a quote for
=> 54	Side	Y	The side to request a quote for. Valid values: - 1 = Buy, - 2 = Sell
=> 38	OrderQty	Y	The quantity to request a quote for
	<MessageTrailer>	Y	

2.4.2 Quote <S>

The Quote message is issued in response to a RFQ to show B2C2's bid/offer prices in the request size.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = S
117	QuoteID	Y	The unique identifier for this quote
131	QuoteReqID	Y	Refers to the unique identifier from the initial request.
55	Symbol	Y	The symbol to request a quote for
54	Side	Y	The side to request a quote for. Valid values: • 1 = Buy, • 2 = Sell
132	BidPx	C	Conditionally required if Side = Sell, this price is what B2C2 bids you
133	OfferPx	C	Conditionally required if Side = Buy, this price is what B2C2 offers you
134	BidSize	C	Conditionally required if Side = Sell, this is the quantity B2C2 bids you
135	OfferSize	C	Conditionally required if Side = Buy, this is the quantity B2C2 offers you
62	ValidUntilTime	Y	The time when the quote will expire, it may expire earlier due to a market move / other factors
60	TransactTime	N	The time the quote was generated
	<MessageTrailer>	Y	

2.4.3 Quote Request Reject <AG>

The Quote Request Reject message is issued to reject a RFQ due to invalid information or technical issues.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = AG
131	QuoteReqID	Y	Refers to the unique identifier from the initial request.
658	QuoteRequestRejectReason	Y	The reason the quote request was rejected.
146	NoRelatedSym	Y	The number of symbols to request a quote for. Valid value: 1
=> 55	Symbol	Y	The symbol the quote request was for
=> 54	Side	Y	The side the quote request was for. Valid values: • 1 = Buy, • 2 = Sell
=> 38	OrderQty	Y	The quantity the quote request was for
58	Text	N	The reason the quote was rejected
	<MessageTrailer>	Y	

2.5 Trade Execution - Requests and Responses

Trades can be executed either on a request for quote (RFQ) basis where a quote is requested in a given size and a price is returned, or through Fill or Kill (FOK) limit orders whereby for a given size the client can submit a price that they wish to trade at or better than.

2.5.1 New Order Single <D> - Executing on an RFQ

The New Order Single message is used to submit orders to B2C2 for execution of previous Quote messages.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = D
11	ClOrdID	Y	The unique identifier assigned by the client to this trade request / order
1	Account	N	An identifier of the client's account for which this order is destined
117	QuoteID	Y	The unique identifier of the Quote
55	Symbol	Y	The symbol of the Quote being executed
54	Side	Y	The side of the Quote being executed
38	OrderQty	Y	The quantity of the Quote being executed
44	Price	Y	The price of the Quote being executed
40	OrdType	Y	Indicates the order type. Valid value: E = Previously indicated
59	TimeInForce	Y	Specified how long the order remains in effect. Valid values: • 3 = Immediate or Cancel (IOC), • 4 = Fill or Kill (FOK)
60	TransactTime	Y	Time of execution/order creation (expressed in UTC)
	<MessageTrailer>	Y	

2.5.2 New Order Single <D> - Executing on a FOK order

The New Order Single message is used to submit a FOK order, which is either immediately executed in full or rejected. The executed price may be better than the price requested on the FOK.

The effective time tag can be specified to set an expiry time on the FOK order to protect against severe latency / accidental resubmission of an order.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = D
11	ClOrdID	Y	The unique identifier assigned by the client for this order
1	Account	N	An identifier of the client's account for which this order is destined
55	Symbol	Y	The symbol of the order
54	Side	Y	The side of the order
38	OrderQty	Y	The quantity of the order
44	Price	Y	The requested price to trade at or better than
40	OrdType	Y	Indicates the order type. Valid value: 2 = Limit
59	TimeInForce	Y	Specified how long the order remains in effect. Valid value: 4 = Fill or Kill (FOK)
60	TransactTime	Y	Time of execution/order creation (expressed in UTC)
168	EffectiveTime	N	The time at which to expire this FOK, maximum is 20 seconds from the current time in UTC
	<MessageTrailer>	Y	

2.5.3 Execution Report <8>

The Execution Report message is issued in response to a New Order Single message, and sent as a duplicate to any associated drop copy sessions. It is used to show the traded quantity of successful order requests and rejected requests.

For CFD trades a single new order may result in a single execution report, or multiple execution reports if the order caused multiple CFD contracts to close. This is entirely configurable on a client by client basis. The default behaviour is a single execution report.

Tag	Name	Req	Description
	<MessageHeader>	Y	MsgType <35> = 8
37	OrderID	Y	The unique identifier assigned by B2C2 to the order chain
11	ClOrdID	Y	Refers to the ClOrdID specified on the initial order
17	ExecID	Y	The unique identifier assigned by B2C2 to each Execution Report message
150	ExecType	Y	Describes the purpose of this execution report. Valid values: • F = Trade, • 8 = Rejected
39	OrdStatus	Y	Describes the current order status. Valid values: • 2 = Filled, • 8 = Rejected
1	Account	C	An identifier of the client’s account, echoed from the incoming order. Present if specified on the originating order
55	Symbol	Y	The symbol the order is for
54	Side	Y	The side the order is for
44	Price	C	The price specified on the initial order
59	TimeInForce	N	Indicates how long the initial order remains in effect
6	AvgPx	Y	The average price of fills on this order.

			Zero for rejected orders
38	OrderQty	Y	The initial order quantity
32	LastQty	C	The total quantity filled. Present when ExecType = Trade(F)
31	LastPx	C	The executed price of this ExecutionReport. Present when ExecType = Trade(F)
14	CumQty	Y	The total quantity filled
151	LeavesQty	Y	The total quantity remaining to be filled. Zero for rejected orders
578	TradeInputSource	C	The source of the trade. May be present on drop copy feed only. Valid values: • "API" = Executed via an API • "Manual" = Executed manually, i.e. via a UI
60	TransactTime	N	Time the transaction represented by this ExecutionReport occurred (expressed in UTC)
58	Text	C	Contains the reject reason. Present when OrdStatus = Rejected(8)
	<MessageTrailer>	Y	

2.6 Drop Copy

Upon a successful log in, the client will receive all trade executions since the start of the current Business Day. This will be immediately followed by any new executions in near real time. The drop copy feed does not support retrieval of trades from previous Business Days.

A drop copy session may not be used for entering orders or requesting market data. All unsupported messages from clients will be rejected and may result in a session disconnection.

B2C2 does not support any type of subscription, nor any acknowledgement of the execution reports.

2.6.1 Message Format

An Execution Report will be sent for every trade, matching the corresponding messages sent via the trading session, with the following differences:

- Price, as in the price of the original order, will not be present but LastPx, the executed price, will be
- ClOrdID may not be present
- OrdStatus is always 2(Filled)
- OrderQty is always 0
- CumQty is always 0
- LeavesQty is always 0
- ExecType is always F(Trade)
- TradeInputSource may be present

2.6.2 Unique Trade Identifiers

The ExecID is the unique identifier for each B2C2 trade. It can be used to reconcile trades with executions that are received on the client's trading FIX sessions. We recommend that clients use this tag as the unique identifier for each trade.

3 Custom Fields and Messages

The implementation of the following custom fields is optional.

3.1 No Quantity Levels <8000>

- Type: NumInGroup
- Description: The number of Quantity <53> entries in a repeating group
- Used In: NoReleasedSym <146> of a Market Data Request <V> message
- Purpose: When subscribing to market data, this field allows the user to specify the different layers of quantity that they want prices for.

4 Example Messages

4.1 Market Data

You can either subscribe to our default levels, or request two levels in a quantity of your own choosing.

Example One - subscribing to default quantity levels:
You do not need to specify tag 8000, if you don’t you will get subscribed to two price levels for approximately $1000 and $5000

Client Sends:

```
8=FIX.4.4|9=167|35=V|34=3|49=COMPID|52=20180313-12:01:15.022|56=B2C2UMD|
262=MDRQ-1520942475018|263=1|264=0|265=0|146=1|55=BTCEUR.SPOT|267=2|
269=0|269=1|10=064|
```

B2C2 Response:

```
8=FIX.4.4|9=235|35=W|34=9|49=B2C2UMD|52=20180313-12:01:18.796|56=COMPID|
55=BTCEUR.SPOT|262=MDRQ-1520942475018|268=4|269=1|270=7315.23|271=0.136|
269=1|270=7315.23|271=0.68|269=0|270=7305.86|271=0.136|269=0|270=7305.49|
271=0.68|10=236|
```

Example Two - subscribing to 7.5 BTC and 10 BTC quantity levels
However if you do specify tag 8000, then you will need to specify tag 53 in the repeating group to specify the quantity levels you are interested in.

Client Sends:

```
8=FIX.4.4|9=187|35=V|34=3|49=COMPID|52=20180313-12:00:11.911|56=B2C2UMD|
262=MDRQ-1520942411902|263=1|264=0|265=0|146=1|55=BTCEUR.SPOT|8000=2|
53=7.5|53=10|267=2|269=0|269=1|10=187|
```

B2C2 Response:

```
8=FIX.4.4|9=232|35=W|34=4|49=B2C2UMD|52=20180313-12:00:13.933|56=COMPID|
55=BTCEUR.SPOT|262=MDRQ-1520942411902|268=4|269=1|270=7342.42|271=7.5|
269=1|270=7344.41|271=10|269=0|270=7326.78|271=7.5|269=0|270=7324.8|271=10|
10=084|
```

4.2 Trading

Quote Request + Execution

Client Sends:

```
8=FIX.4.4|9=162|35=R|34=2|49=COMPID|52=20180314-09:38:10.433|56=B2C2UT|
131=296b3228-8c0a-4f91-b480-03b287cc828d|146=1|55=BTCEUR.SPOT|54=2|
38=0.11|10=029|
```

B2C2 Response:

```
8=FIX.4.4|9=235|35=S|34=2|49=B2C2UT|52=20180314-09:38:10.490|56=COMPID|54=2|
55=BTCEUR.SPOT|62=20180314-09:38:30.000|117=5d52290c-3d35-41aa-9ef3-
1b9a0b16054d|131=296b3228-8c0a-4f91-b480-03b287cc828d|132=7223.15|134=0.11|
10=083|
```

Example One

You send a NewOrderSingle to execute on the quote

Client Sends:

```
8=FIX.4.4|9=223|35=D|34=3|49=COMPID|52=20180314-09:38:13.356|56=B2C2UT|
11=TREQ1521020293353|38=0.11|40=D|44=7223.15|54=2|55=BTCEUR.SPOT|59=3|
60=20180314-09:38:13.353|117=5d52290c-3d35-41aa-9ef3-1b9a0b16054d|10=113|
```

B2C2 Response:

```
8=FIX.4.4|9=287|35=8|34=3|49=B2C2UT|52=20180314-09:38:15.077|56=COMPID|
6=7223.15|11=TREQ1521020293353|14=0.11|17=a835fc10-6dcc-47cd-a0a0-
5081ffda273d|37=a835fc10-6dcc-47cd-a0a0-5081ffda273d|38=0.11|39=2|44=7223.15|
54=2|55=BTCEUR.SPOT|60=20180314-09:38:13.000|150=F|151=0|10=254|
```

Example Two - Fill or Kill order

You send FOK NewOrderSingle Buy 1.2 BTCEUR.SPOT 9000

Client Sends:

```
8=FIX.4.4|9=197|35=D|34=2|49=COMPID|52=20180314-09:49:16.786|56=B2C2UT|
11=6867488e-5e0f-42b6-b95a-7043070e1b51|38=1.2|40=2|44=9000|54=1|
55=BTCEUR.SPOT|59=4|60=20180314-09:49:16.783|10=153|
```

We reply with execution report (you will see that you were filled at 7284.3321, so got a price improvement on your FOK)

B2C2 Response:

```
8=FIX.4.4|9=303|35=8|34=2|49=B2C2UT|52=20180314-09:49:17.751|56=COMPID|
6=7284.3321|11=6867488e-5e0f-42b6-b95a-7043070e1b51|14=1.2|17=97674e15-
5424-4994-8df8-7a621b0edd53|37=97674e15-5424-4994-8df8-7a621b0edd53|
38=1.2|39=2|44=9000|54=1|55=BTCEUR.SPOT|60=20180314-09:49:17.000|150=F|
151=0|10=009|
```

Example Three - Fill or Kill order for CFDs when client is configured for multiple executions.
For now all orders will be fully filled, or rejected in full.

You can receive partial execution reports, but this is only for CFD trading.

Let’s say you have no open CFD positions and do the following trades:

1.Buy 1 BTCUSD Cfd, you will get a single execution report

2. Buy 1 BTCUSD Cfd, you will get a single execution report

3. Sell 3 BTCUSD Cfd, you will get three execution reports, 2 partial fills that close out 1) and 2) and then a fill for the rest of your order

```
Client Sends:
8=FIX.4.4|9=196|35=D|34=2|49=COMPID|52=20180601-09:22:45.748|56=B2C2UT|
11=821063e4-4664-4b45-9972-3ccd13a81de5|38=1|40=2|44=9000|54=1|
55=BTCUSD.CFD|59=4|60=20180601-09:22:45.747|10=084|

B2C2 Response:
8=FIX.4.4|9=298|35=8|34=2|49=B2C2UT|52=20180601-09:22:46.240|56=COMPID|
6=7588.72|11=821063e4-4664-4b45-9972-3ccd13a81de5|14=1|17=d4b2a58e-f5fc-
470a-b012-4babd7df7d21|37=d4b2a58e-f5fc-470a-b012-4babd7df7d21|38=1|39=2|
44=9000|54=1|55=BTCUSD.CFD|60=20180601-09:22:46.000|150=F|151=0|10=198|

Client Sends:
8=FIX.4.4|9=196|35=D|34=3|49=COMPID|52=20180601-09:22:50.750|56=B2C2UT|
11=afdc610b-e9e1-40f3-9de1-ce35dde31e68|38=1|40=2|44=9000|54=1|
55=BTCUSD.CFD|59=4|60=20180601-09:22:50.749|10=195|

B2C2 Response:
8=FIX.4.4|9=298|35=8|34=3|49=B2C2UT|52=20180601-09:22:51.075|56=COMPID|
6=7589.21|11=afdc610b-e9e1-40f3-9de1-ce35dde31e68|14=1|17=6f202bb9-27e9-
4ad0-9465-f637bc4ba6f2|37=6f202bb9-27e9-4ad0-9465-f637bc4ba6f2|38=1|39=2|
44=9000|54=1|55=BTCUSD.CFD|60=20180601-09:22:50.000|150=F|151=0|10=228|

Client Sends:
8=FIX.4.4|9=196|35=D|34=4|49=COMPID|52=20180601-09:22:55.752|56=B2C2UT|
11=9b56f5db-fd58-4753-99fb-f9064c03484e|38=3|40=2|44=6000|54=2|
55=BTCUSD.CFD|59=4|60=20180601-09:22:55.751|10=249|

B2C2 Response:
```

```
8=FIX.4.4|9=298|35=8|34=4|49=B2C2UT|52=20180601-09:22:56.254|56=COMPID|
6=7573.89|11=9b56f5db-fd58-4753-99fb-f9064c03484e|14=1|17=3d85b81a-6c37-
4db7-a066-98d6da5f30d9|37=3d85b81a-6c37-4db7-a066-98d6da5f30d9|38=1|39=1|
44=6000|54=2|55=BTCUSD.CFD|60=20180601-09:22:56.000|150=F|151=2|10=051|
```

B2C2 Response:

```
8=FIX.4.4|9=298|35=8|34=5|49=B2C2UT|52=20180601-09:22:56.260|56=COMPID|
6=7573.89|11=9b56f5db-fd58-4753-99fb-f9064c03484e|14=2|17=c61e7ead-02d2-
4644-ae8e-684923a1af9c|37=c61e7ead-02d2-4644-ae8e-684923a1af9c|38=1|39=1|
44=6000|54=2|55=BTCUSD.CFD|60=20180601-09:22:56.000|150=F|151=1|10=125|
```

B2C2 Response:

```
8=FIX.4.4|9=298|35=8|34=6|49=B2C2UT|52=20180601-09:22:56.264|56=COMPID|
6=7573.89|11=9b56f5db-fd58-4753-99fb-f9064c03484e|14=3|17=ee20a8c6-5b58-
48d1-aa74-dc9ae1a86fea|37=ee20a8c6-5b58-48d1-aa74-dc9ae1a86fea|38=1|39=2|
44=6000|54=2|55=BTCUSD.CFD|60=20180601-09:22:56.000|150=F|151=0|10=163|
```