

---

# **FXCM API**

***Release 1.0***

**Forex Capital Markets**

**Jun 12, 2019**



<b>1</b>	<b>REST API Specifications</b>	<b>3</b>
1.1	Getting Started . . . . .	3
1.2	Connect . . . . .	4
1.3	Subscribe vs Snapshot . . . . .	5
1.4	OrderID vs TradeID . . . . .	5
1.5	Candle Request Limit . . . . .	6
1.6	Placing Trailing Stops . . . . .	6
1.7	Account Name vs Account ID . . . . .	6
1.8	Offer Ids . . . . .	7
<b>2</b>	<b>FXCMPY Python Wrapper</b>	<b>11</b>
2.1	Demo Account . . . . .	11
2.2	Package Installation . . . . .	11
2.3	API Token . . . . .	11
2.4	Configuration file . . . . .	12
2.5	Documentation . . . . .	12
2.6	fxcmpy wrapper Sample Code . . . . .	12
<b>3</b>	<b>REST Sample Codes</b>	<b>15</b>
3.1	Connecting . . . . .	15
3.2	Backtest and Strategy Sample Codes . . . . .	15
3.3	PHP . . . . .	15
3.3.1	Requirements . . . . .	15
3.3.2	Installation . . . . .	16
3.3.3	Usage . . . . .	16
3.3.4	Configuration Parameters . . . . .	16
3.3.5	Functions . . . . .	16
3.3.6	Signals . . . . .	17
3.3.7	PHP Sample Code . . . . .	17
3.4	Python . . . . .	18
3.4.1	Getting started . . . . .	18
3.4.2	Using Jupyter Notebook . . . . .	18
3.4.3	Details . . . . .	18
3.4.4	Candles . . . . .	19
<b>4</b>	<b>FIX API Specifications</b>	<b>23</b>
4.1	Getting Started . . . . .	23

4.2	Connecting . . . . .	24
4.3	Marketdata . . . . .	24
4.4	Table . . . . .	25
4.5	Order . . . . .	26
4.6	FIX Sample Solutions and Projects . . . . .	26
4.6.1	C++ FIX Example . . . . .	26
4.6.2	C++ QuickFIX Example VS2017 with x64 . . . . .	26
4.6.3	C# .NET FIX Example . . . . .	27
4.6.4	Java FIX Example . . . . .	27
4.6.5	C++ LAMM Trading Example . . . . .	27
<b>5</b>	<b>FIX Core Concepts</b>	<b>29</b>
5.1	TradingSessionStatusRequest (g) . . . . .	29
5.2	TradingSessionStatus Message (h) . . . . .	29
5.3	CollateralInquiry (BB) . . . . .	30
5.4	Requesting Market Data . . . . .	30
5.5	Getting Positions . . . . .	31
5.5.1	Open Position vs. Closed Position . . . . .	31
5.5.1.1	Open Positions . . . . .	31
5.5.1.2	Closed Positions . . . . .	31
5.5.1.3	Position Margin . . . . .	31
5.6	Overview of Basic Order and Time-In-Force Types . . . . .	32
5.6.1	Time-In-Force (TIF) Types . . . . .	32
5.6.1.1	Good Til Cancel (GTC) . . . . .	32
5.6.1.2	Day . . . . .	32
5.6.1.3	Immediate or Cancel (IOC) . . . . .	32
5.6.1.4	Fill or Kill (FOK) . . . . .	32
5.6.2	Order Types . . . . .	32
5.6.2.1	Market . . . . .	32
5.6.2.2	Market Range (Stop-Limit) . . . . .	33
5.6.2.3	Limit . . . . .	33
5.6.2.4	GTC/Day Limit Order . . . . .	33
5.6.2.5	IOC/FOK Limit Order . . . . .	34
5.6.2.6	Stop . . . . .	34
5.6.2.7	Trailing Stop Peg Order . . . . .	34
5.7	Handling of Partial Fills . . . . .	38
5.7.1	Order Quantity Fields . . . . .	38
5.7.2	Example Partial Fill . . . . .	39
5.7.3	Closing A Position . . . . .	39
5.7.4	Sending A Closing Order . . . . .	40
5.8	Getting Account Position Maintenance . . . . .	40
5.8.1	Getting Position Maintenance in Code . . . . .	40
5.9	When To Reset MsgSeqNum . . . . .	41
5.9.1	Reset On Logon . . . . .	41
5.10	Account Equity . . . . .	41
5.11	StartCash(921) . . . . .	41
5.12	Steps To Retrieve Short Version Of Market Price . . . . .	41
5.13	EMF . . . . .	41
5.14	Execution Disclaimer . . . . .	42
<b>6</b>	<b>Frequently Asked Questions</b>	<b>43</b>
6.1	Error Messages received . . . . .	43
6.2	Limit order Day order vs Limit IOC/FOK . . . . .	44
6.3	What is the Base Unit Size (also the minimum trade size) for all FX instruments? . . . . .	45

6.4	How can I get Positions side and quantity from open position report? . . . . .	45
6.5	How can I get closed positions? . . . . .	45
6.6	How can I close open positions? . . . . .	46
6.7	Sample Code in C++ . . . . .	46
6.8	FXCM Custom fields . . . . .	53
6.8.1	From TradingSessionStatus(h) . . . . .	53
6.8.2	From MarketDataSnapshotFullRefresh(W) . . . . .	54
6.8.3	From CollateralReport(BA) . . . . .	54
6.9	FXCM System Parameters . . . . .	55
<b>7</b>	<b>JAVA API Specifications</b>	<b>57</b>
7.1	Getting Started . . . . .	57
7.2	Running QATest.java . . . . .	57
7.3	Login . . . . .	58
7.4	Rollover . . . . .	59
7.5	Sample Codes . . . . .	59
<b>8</b>	<b>ForexConnect API Specifications</b>	<b>61</b>
8.1	Getting Started . . . . .	61
8.2	Top Development Platform IDEs . . . . .	61
8.3	Table manager vs Non-table manager . . . . .	62
8.4	Request Current Balance . . . . .	62
8.5	Retrieve Price History . . . . .	62
8.6	Sample Codes . . . . .	63
<b>9</b>	<b>Market Data</b>	<b>65</b>
9.1	TickData . . . . .	65
9.1.1	Sample tick data code . . . . .	66
9.1.1.1	TickData34.py . . . . .	66
9.1.1.2	TickData27.py . . . . .	67
9.2	CandleData . . . . .	68
9.2.1	Sample candle data code . . . . .	68
9.2.1.1	CandleData(pandas).py . . . . .	68
9.2.1.2	CandleData34.py . . . . .	69
9.2.1.3	CandleData27.py . . . . .	70
9.3	Order Flow . . . . .	71
9.4	Sentiment . . . . .	72
9.5	Volume . . . . .	72
<b>10</b>	<b>Support</b>	<b>75</b>
<b>11</b>	<b>License</b>	<b>77</b>



**Algorithmic Trading APIs for Forex and CFDs:**

FXCM offers APIs ideal to automate your trading strategies. Learn about our REST API, FIX, JAVA and ForexConnect.





---

## REST API Specifications

---

Our **REST API** is a web-based API using a Websocket connection and was developed with algorithmic trading in mind.

Developers and investors can create custom trading applications, integrate into our platform, back test strategies and build robot trading. Calls can be made in any language that supports a standard HTTP.

We utilize the new OAuth 2.0 specification for authentication via token. This allows for a more secure authorization to access your application and can easily be integrated with web applications, mobile devices, and desktop platforms

With the use of the socket.io library, the API has streaming capability and will push data notifications in a JSON format. Your application will have access to our real-time streaming market data, subscribe in real time access to trading tables and place live trades.

### **FXCM Trading hours**

FXCM's trading hours vary by product. For forex, trading opens on Sundays between 5:00 PM ET and 5:15 PM ET and closes on Fridays around 4:55 PM ET. For CFDs, please check the [CFD Product Guide](#).

## 1.1 Getting Started

### 1. Quick start guide

- [Python](#)
- [Node.js](#)
- [Java](#)

### 2. Sample code

- [Node.js sample code](#)
- [Java sample code](#)
- [C# sample code](#)

### 3. Account setup

- Apply for a [demo account](#).
  - Generate access token. You can generate one from [Trading Station web](#). Click on User Account > Token Management on the upper right hand of the website.
  - For live account, please send your username to [api@fxcm.com](mailto:api@fxcm.com), we will need to enable Rest API access. For demo account, Rest API access is enabled by default.
4. Download Rest API document.
  5. Access REST API documentation using [Swagger](#).
  6. Start coding. You will need to reference the [socket.io library](#) in your code.
    - [Socket.io for Javascript](#)
    - [Socket.io for Python](#)

## 1.2 Connect

Clients should establish a persistent WebSocket connection using socket.io library. All non-solicited updates will be sent over this connection. Client requests are to be sent via normal HTTP messages. Every HTTP message must contain following parameters:

Table 1: HTTP Message

HTTP Message	
Header	Description
HTTP version	Version of HTTP used
User-Agent	Identification of the client software
Accept	Acceptable response MIME type
Content-Type	Media type of the request
Authorization	<b>Authorization string containing “Bearer “, ID of socket tion and persistent token</b>

Sample Request:

```
GET/socket.io/?access_token=cj5wedhq3007v61fe935ihqed&EIO=3&transport=polling&t=Lsd_
↪lZY&b64=1
HTTP/1.1
User-Agent: node-XMLHttpRequest
Accept: */*
Host: api.fxcm.com
Connection: close
```

What ‘t’ means:

"t" is the table id:

Table 2: TableID

TableID	
t	table ID
1	Open Position

Continued on next page

Table 2 – continued from previous page

TableID	
2	Closed Position
3	Order
4	OCO Orders
5	Summary
6	Account

## 1.3 Subscribe vs Snapshot

FXCM Rest API provides two ways to deliver data. subscribe vs snapshot.

After subscribe, data will be pushed to your socket whenever there is an update. You can subscribe Market data stream /subscribe or live table update /trading/subscribe. You can also unsubscribe. You can request a snapshot of trading tables via /trading/get\_model.

```
Model choices: 'Offer', 'OpenPosition', 'ClosedPosition', 'Order', 'Summary',
↳ 'LeverageProfile', 'Account', 'Properties'.
```

## 1.4 OrderID vs TradeID

OrderID and TradeID are different. In Market order, an order id is created straightaway and it is in callback immediately.

```
{"response":{"executed":true},"data":{"type":0,"orderId":81712802}}
```

A trade id is not generated until after order is executed. You have to subscribe the order table and listing the live update and look up the trade id. You will not get a trade id in snapshot as the information disappears when you submit the request.

```
Examples:
Subscribing for Orders table:
POST /trading/subscribe
models=Order

Placing Market order:
POST /trading/open_trade
account_id=1537581&symbol=EUR%2FUSD&is_buy=false&rate=0&amount=5&order_type=AtMarket&
↳ time_in_force=GTC

Response from server:
{"executed":true>{"type":0,"orderId":390285837}

Received Order record from /trading/subscribe with order_id and trade_id:
{"t":3,"ratePrecision":5,"orderId":"390285837","tradeId":"170162801","time":
↳ "04252018120716391","accountName":"01537581","accountId":"1537581","timeInForce":
↳ "GTC","expireDate":"","currency":"EUR/USD","isBuy":false,"buy":0,"sell":1.21818,
↳ "type":"OM","status":2,"amountK":5,"currencyPoint":0.5,"stopMove":0,"stop":0,
↳ "stopRate":0,"limit":0,"limitRate":0,"isEntryOrder":false,"ocoBulkId":0,
↳ "isNetQuantity":false,"isLimitOrder":false,"isStopOrder":false,"isELSOOrder":false,
↳ "stopPegBaseType":-1,"limitPegBaseType":-1,"range":0,"action":"I"}
```

Furthermore, a single market order can have many TradeIDs, if they are partial fills or closing of other orders. In this case, it's more appropriate to provide the OrderID which ties back to that specific market order request, from there you can join this OrderID to any associated order.

In an entry order, an order ID is in callback function. You can also see it on an order table snapshot. but you will not get a TradeID until order been executed.

## 1.5 Candle Request Limit

Table 3: Candle download limit

Download limits		
Time-frame	Max Days Back	Max Num
m1	16	10,000
m5	56	10,000
m15	212	10,000
m30	316	10,000
h1	624	10,000
h2	1224	10,000
h3	2056	10,000
h4	2664	10,000
h6	3632	10,000
h8	5128	10,000
D1, W1, M1	no limit	no limit

## 1.6 Placing Trailing Stops

The fixed trailing stop should be 10 or above, for dynamic trailing stop = 1, number between 2-9 will be rejected. Parameter is trailing\_stop\_step.

```
Example Entry order with trailing stop of 10 pips:
POST /trading/create_entry_order account_id=1537581&symbol=EUR%2FUSD&is_buy=true&
↪rate=1.1655&amount=3&order_type=Entry&time_in_force=GTC&stop=-50&trailing_stop_
↪step=10&is_in_pips=true
```

## 1.7 Account Name vs Account ID

There is a difference between account name and account id. Usually removing the heading zeros are account ID. You need to pass the account\_id when placing orders. You can retrieve this information from /trading/get\_model/accounts.

```
Wrong:
{"is_buy":false,"account_id":"00654061","symbol":"EUR/USD","rate":1.15,"amount":11,
↪"stop":-40,"is_in_pips":true,"order_type":"AtMarket","time_in_force":"GTC"}

ERR noExec: /trading/create_entry_order
{"code":3,"message":"Amount should be divisible by 10","parameters":["10"]}
```

(continues on next page)

(continued from previous page)

Correct:

```
{ "is_buy": false, "account_id": "654061", "symbol": "EUR/USD", "rate": 1.15, "amount": 11, "stop
↪": -40, "is_in_pips": true, "order_type": "AtMarket", "time_in_force": "GTC" }
```

```
request # 2 has been executed: {
"response": { "executed": true }, "data": { "type": 0, "orderId": 194963057 } }
```

## 1.8 Offer Ids

Table 4: Offer Ids

Symbol	OfferId
EUR/USD	1
USD/JPY	2
GBP/USD	3
USD/CHF	4
EUR/CHF	5
AUD/USD	6
USD/CAD	7
NZD/USD	8
EUR/GBP	9
EUR/JPY	10
GBP/JPY	11
CHF/JPY	12
GBP/CHF	13
EUR/AUD	14
EUR/CAD	15
AUD/CAD	16
AUD/JPY	17
CAD/JPY	18
NZD/JPY	19
GBP/CAD	20
GBP/NZD	21
GBP/AUD	22
AUD/NZD	28
USD/SEK	30
EUR/SEK	32
EUR/NOK	36
USD/NOK	37
USD/MXN	38
AUD/CHF	39
EUR/NZD	40
USD/ZAR	47
USD/HKD	50
ZAR/JPY	71
USD/TRY	83
EUR/TRY	87
NZD/CHF	89

Continued on next page

Table 4 – continued from previous page

Symbol	OfferId
CAD/CHF	90
NZD/CAD	91
TRY/JPY	98
USD/ILS	100
USD/CNH	105
AUS200	1001
ESP35	1002
FRA40	1003
GER30	1004
HKG33	1005
JPN225	1007
NAS100	1008
SPX500	1010
UK100	1012
US30	1013
Copper	1016
CHN50	1020
EUSTX50	1035
USDOLLAR	1058
US2000	1060
USOil	2001
UKOil	2002
SOYF	2003
NGAS	2015
WHEATF	2020
CORNF	2021
Bund	3001
XAU/USD	4001
XAG/USD	4002
XAU/EUR	4005
XAU/AUD	4006
XAU/TRY	4007
ETH/USD	8003
LTC/USD	8004

**Note:** This is for personal use and abides by our [EULA](#). For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com)

**Disclaimer:**

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM

Australia Pty. Limited (“FXCM AU”) (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group’s websites prior to taking further action.”





---

### FXCMPY Python Wrapper

---

FXCM provides a RESTful API to interact with its trading platform. Among others, it allows the retrieval of historical data as well as of streaming data. In addition, it allows to place different types of orders and to read out account information. The overall goal is to allow the implementation automated, algorithmic trading programs. `fxcmpy.py` is a Python wrapper package for that API.

## 2.1 Demo Account

To get started with the the API and the package, a demo account with FXCM is sufficient. You can open such an account under <https://www.fxcm.com/uk/forex-trading-demo/>.

## 2.2 Package Installation

Installation happens via `pip` install on the command line.

```
pip install fxcmpy
```

## 2.3 API Token

To connect to the API, you need an API token that you can create or revoke from within your (demo) account in [Trading Station](#).

In an interactive context, you can use e.g. a variable called `TOKEN` to reference your unique API token.

```
TOKEN = YOUR_FXCM_API_TOKEN
Connecting to the server, then boils down to the following line of code.

con = fxcmpy.fxcmpy(access_token=TOKEN, log_level='error')
```

## 2.4 Configuration file

It is recommended to store the API token in a configuration file which allows for re-usability and hides the token on the GUI level. The file should contain the following lines.

```
[FXCM]
log_level = error
log_file = PATH_TO_AND_NAME_OF_LOG_FILE
access_token = YOUR_FXCM_API_TOKEN
```

It is assumed onwards that this file is in the current working directory and that its name is `fxcm.cfg`.

With such a configuration file in the current working directory, only the filename need to be passed as a parameter to connect to the API.

```
con = fxcmpy.fxcmpy(config_file='fxcm.cfg')
```

## 2.5 Documentation

- The detailed documentation of this wrapper is found under:

<https://www.fxcm.com/fxcmpy/>

- The detailed documentation of the API is found under:

<https://github.com/fxcm/RestAPI>

## 2.6 fxcmpy wrapper Sample Code

1. EMA Crossover Strategy.ipynb
2. Real-Time SMA Crossover Strategy.ipynb
3. EMA Crossover Strategy and Backtest-checkpoint.ipynb
4. Hull Moving Average Strategy.ipynb
5. Bollinger Band Backtest Part 1.ipynb
6. Bollinger Band Backtest Part 2.ipynb
7. EMA Crossover Strategy backtest.py
8. Introduction to Monte Carlo Simulation.ipynb
9. Stochastic Strategy Backtest.py
10. BB ADX Range Strategy.py
11. Bitcoin Breakout Strategy.py
12. Python Strategy Template.py
13. RSI Range Strategy with Trading Time Range.py
14. RSI Range Strategy.py
15. RSI with SMA Trend Filter.py
16. SMA Crossover Strategy.py

---

**Note:** This is for personal use and abides by our [EULA](#). For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com)

---

### **Disclaimer**

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action.



### 3.1 Connecting

1. [FXCM REST API Demonstration](#)
2. [Quick Start.ipynb](#)
3. [Connecting.ipynb](#)

### 3.2 Backtest and Strategy Sample Codes

1. `bt` backtest using FXCM historical data. What is `bt`?
2. `QSTrader` using FXCM data. What is `QSTrader`?
3. RSI Strategy
4. Moving Average Crossover Strategy
5. [Video demonstration](#) on how to backtest strategies in Visual Studio using FXCM data on QuantConnect LEAN platform.

### 3.3 PHP

`FxcmRest` is a library for event-driven trading with FXCM over RestAPI using `ReactPHP`.

#### 3.3.1 Requirements

PHP 7.0.2+

### 3.3.2 Installation

The recommended way to install FxcmRest is through [Composer](#).

This command will install the latest stable version:

```
$ composer require fxcm/fxcmrest
```

### 3.3.3 Usage

As FXCM Rest API requires you to keep Socket.IO connection open through the whole time it is used, this library must be run within a php script and not as part of php generated website.

Interaction can be done either by using a console or through HTTP requests handled directly by the php script for example with `\React\HTTP`.

Main class of the library is `\FxcmRest\FxcmRest`. It must be instantiated with two objects:

- `\React\EventLoop\LoopInterface`
- `\FxcmRest\Config`

Configuration class `\FxcmRest\Config` must be instantiated with an array containing at least the two following parameters:

- `host`
- `token`

### 3.3.4 Configuration Parameters

- `protocol` - either `\FxcmRest\Protocol::HTTPS` (default) or `\FxcmRest\Protocol::HTTP`
- `host` - either `api.fxcm.com` for Real accounts or `api-demo.fxcm.com` for Demo accounts
- `port` - port number. 443 default
- `token` - 40 char hexadecimal string

### 3.3.5 Functions

```
connect() : null
```

1. Opens a connection to the server. When connection is complete, `connected` signal will be emitted.

```
disconnect() : null
```

2. Disconnects from the server. When disconnection is complete, `disconnected` signal will be emitted.

```
socketID() : string
```

3. If connected to the server, returns a string representing the `socketID`. If not connected, returns an empty string.

```
request(\FxcmRest\HttpMethod $method, string $path, array $arguments, callable  
↪$callback) : null
```

4. Sends a http request to the server. When request is completed, `$callback` will be called with two parameters:

- int representing HTTP status code. 200 for OK
- string representing server answer body

```
on(string $signalName, callable $callback) : null
```

5. Registers a \$callback for a signal of \$signalName. For a list of signals and parameters that are passed with them please see **Signals** section.

### 3.3.6 Signals

1. connected - Emitted when connection sequence is complete. After this socketID is valid and requests can be sent to the server. No parameters are passed.
2. disconnected - Emitted when connection to the server is closed. No parameters are passed.
3. error - Emitted on errors. Passes error description as string.
4. [Offer, OpenPosition, ClosedPosition, Account, Summary, Properties] - Emitted on trading table changes. Passes table update contents as JSON string. Requires subscription through /trading/subscribe
5. (EUR/USD, EUR/GBP, ...) - Emitted on price update. Passes the price update as a JSON string. Requires subscription through /subscribe.

### 3.3.7 PHP Sample Code

```
<?php
require_once __DIR__ . '/vendor/autoload.php';

$loop = \React\EventLoop\Factory::create();

$config = new \FxcMRest\Config([
    'host' => 'api-demo.fxcm.com',
    'token' => 'YOUR_TOKEN',
]);

$counter = 0;
$rest = new \FxcMRest\FxcMRest($loop, $config);
$rest->on('connected', function() use ($rest, &$counter) {
    $rest->request('POST', '/subscribe',
        ['pairs' => 'EUR/USD'],
        function($code, $data) use ($rest, &$counter) {
            if($code === 200) {
                $rest->on('EUR/USD', function($data) use ($rest, &$counter) {
                    echo "price update: {$data}\n";
                    $counter++;
                    if($counter === 5) {
                        $rest->disconnect();
                    }
                });
            }
        });
    }
});

$rest->on('error', function($e) use ($loop) {
```

(continues on next page)

(continued from previous page)

```
    echo "socket error: {$e}\n";
    $loop->stop();
});
$rest->on('disconnected', function() use ($loop) {
    echo "FxcRest disconnected\n";
    $loop->stop();
});
$rest->connect();

$loop->run();
?>
```

## 3.4 Python

---

**Note:** REST API Python code sample - **fxcm-api-rest-python3-example**. Clone this repository by clicking [here](#).

---

### 3.4.1 Getting started

1. Install [Python](#).
2. Run: `pip install -r requirements.txt`
3. Within the `fxcm_rest.json` file:
  - Set log path via the `logpath` field
  - Set `debugLevel` if desired
  - Set subscription lists if desired
4. In the `fxcm_rest_client_sample.py` file:
  - Set your token and environment (demo/real)

### 3.4.2 Using Jupyter Notebook

1. Install [Python](#).
2. Run: `pip install jupyter` (if you don't have jupyter installed already)
3. Run: `pip install -r requirements.txt`
4. In this directory run: `jupyter notebook`
5. Start the `RestApiNotebook.ipynb`.

### 3.4.3 Details

This API exposes the methods of the REST API as a class, dealing with all of the common tasks involved with setting up connections and wiring callback listeners for you. In addition to that there are a few convenience methods.

A quick example is as follows:



```

import fxcm_rest_api_token
import time
trader = fxcm_rest_api_token.Trader('YOURTOKEN', 'prod')
trader.login()

#### Open Market Order
# query account details and use the first account found
accounts = trader.get_model("Account")
account_id = accounts['accounts'][0]['accountId']
# Open 10 lots on USD/JPY for the first account_id found.
response = trader.open_trade(account_id, "USD/JPY", True, 10)
if response['status']:
    # close all USD/JPY trades.
    response = trader.close_all_for_symbol("USD/JPY")

#### Historical Data request
basic = trader.candles("USD/JPY", "m1", 5)
print(basic)
date_fmt = trader.candles("USD/JPY", "m1", 5, dt_fmt="%Y/%m/%d %H:%M:%S")
print(date_fmt)
date_fmt_headers = trader.candles_as_dict("USD/JPY", "m1", 3, dt_fmt="%Y/%m/%d %H:%M:
↪%S")
print(date_fmt_headers)
#### Price subscriptions
subscription_result = trader.subscribe_symbol("USD/JPY")

# Define alternative price update handler and supply that.
def pupdate(msg):
    print("Price update: ", msg)
subscription_result = trader.subscribe_symbol("USD/JPY", pupdate)
counter = 1
while counter < 60:
    time.sleep(1)
    counter += 1

```

### 3.4.4 Candles

All calls to candles allow either `instrument name`, or `offerId`. They also allow the `From` and `To` to be specified as timestamp or a date/time format that will be interpreted (“2017/08/01 10:00”, “Aug 1, 2017 10:00”, etc.). In addition to `instrument_id`, `response`, `period_id` and `candles`, a `headers` field (not documented in the API notes) is returned, representing the candle fields.

basic

```

for item in basic['candles']:
    print item

[1503694500, 109.317, 109.336, 109.336, 109.314, 109.346, 109.366, 109.373, 109.344, ↵
↪72]
[1503694560, 109.336, 109.321, 109.337, 109.317, 109.366, 109.359, 109.366, 109.354, ↵
↪83]
[1503694620, 109.321, 109.326, 109.326, 109.316, 109.359, 109.358, 109.362, 109.357, ↵
↪28]

```

date\_fmt

```
for item in date_fmt['candles']:
    print item

[1503694500, 109.317, 109.336, 109.336, 109.314, 109.346, 109.366, 109.373, 109.344, ↵
↵72, '2017/08/26 05:55:00']
[1503694560, 109.336, 109.321, 109.337, 109.317, 109.366, 109.359, 109.366, 109.354, ↵
↵83, '2017/08/26 05:56:00']
[1503694620, 109.321, 109.326, 109.326, 109.316, 109.359, 109.358, 109.362, 109.357, ↵
↵28, '2017/08/26 05:57:00']
```

date\_fmt\_headers

```
for item in date_fmt_headers['candles']:
    print item

Headers(timestamp=1503694620, bidopen=109.321, bidclose=109.326, bidhigh=109.326, ↵
↵bidlow=109.316, askopen=109.359, askclose=109.358, askhigh=109.362, asklow=109.357, ↵
↵tickqty=28, datestring='2017/08/26 05:57:00')
Headers(timestamp=1503694680, bidopen=109.326, bidclose=109.312, bidhigh=109.326, ↵
↵bidlow=109.31, askopen=109.358, askclose=109.374, askhigh=109.376, asklow=109.358, ↵
↵tickqty=42, datestring='2017/08/26 05:58:00')
Headers(timestamp=1503694740, bidopen=109.312, bidclose=109.312, bidhigh=109.312, ↵
↵bidlow=109.31, askopen=109.374, askclose=109.374, askhigh=109.374, asklow=109.372, ↵
↵tickqty=4, datestring='2017/08/26 05:59:00')

for item in date_fmt_headers['candles']:
    print "%s: Ask Close [%s], High Bid [%s] " % (item.datestring, item.askclose, ↵
↵item.bidhigh)

2017/08/26 05:57:00: Ask Close [109.358], High Bid [109.326]
2017/08/26 05:58:00: Ask Close [109.374], High Bid [109.326]
2017/08/26 05:59:00: Ask Close [109.374], High Bid [109.312]
```

subscribe\_symbol - default

```
{u'Updated': 1504167080, u'Rates': [110.467, 110.488, 110.629, 110.156], u'Symbol': u
↵'USD/JPY'}
{u'Updated': 1504167081, u'Rates': [110.469, 110.49, 110.629, 110.156], u'Symbol': u
↵'USD/JPY'}
```

subscribe\_symbol - overridden

```
Price update: {"Updated":1504167248,"Rates":[110.446,110.468,110.629,110.156],"Symbol
↵":"USD/JPY"}
Price update: {"Updated":1504167250,"Rates":[110.446,110.468,110.629,110.156],"Symbol
↵":"USD/JPY"}
```

---

**Note:** This is for personal use and abides by our [EULA](#). For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com)

---

#### Disclaimer:

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients.

For clients who maintain account(s) with Forex Capital Markets Limited (“FXCM LTD”), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the “FXCM Group”], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited (“FXCM AU”) (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group’s websites prior to taking further action.”



---

## FIX API Specifications

---

### Based on FIX designed for institutional traders

FIX API using FIX Protocol 4.4 is designed for real-time, custom institutional interface which can push up to 200 price update per second (not available on other APIs). It is our fastest and most popular solution. You will get full range of trading order types available at FXCM.

In order to establish and maintain FIX connectivity, you must have an application that manages a network connection and which sends/receives FIX messages. An application that does this is referred to as a FIX engine. Today there are numerous commercial FIX engines as well as open-source alternatives, the most known of which is QuickFIX.

FXCM trading session reset weekly, it opens on Sundays between 5:00 PM ET and 5:15 PM ET. and closes on Fridays at 4:55 PM ET.

Please refer to our [GitHub](#).

## 4.1 Getting Started

- Open a demo [Trading Station II](#) account.
- Send your login to [api@fxcm.com](mailto:api@fxcm.com) to get FIX credentials.
- Request documentation by signing our [EULA](#).
- Download the FXCM data dictionary.
- Install software development environment (SDE). For Java, you can try [Eclipse](#) or [NetBeans](#). For C++/C#, please download [Visual Studio](#).
- You also need to download FIX protocol package. [QuickFix/J](#) or [QuickFix/N](#).
- Download sample QuickFIX configuration file.
- Download the [quickstart](#) guide.

## 4.2 Connecting

After your application has created a FIX session, you can begin sending and receiving FIX messages. However, there is a sequence of messages that should be sent prior to conducting any other messaging activity. These messages are described below.

This is the first message that you must send. Any messages you send before this will be ignored. There are two ways to Logon, one is within the Logon message should include your Username (553) and Password (554). The other one is send Username (553) and Password (554) on User Request (35=BE). It is also important to note here that FXCM requires the TargetSubID on all messages, including your Logon. If you are not receiving any responses to your Logon message, it is likely because you have not included TargetSubID.

### Example Logon Messages

- Send Username/Password on Logon (35=A)

```
8=FIX.4.4|9=114|35=A|34=1|49=fx1294946_client1|52=20120927-13:15:34.754|56=FXCM_
↪|57=U100D1|553=fx1294946|554=123|98=0|108=30|141=Y|10=146|
```

- Send Username/Password on User Request (35=BE)

```
8=FIX.4.4|9=114|35=BE|34=2|49=fx1294946_client1|52=20140515-00:29:11.
↪372|56=FXCM|57=U100D1|553=fx1294946|554=1234|923=1|924=1|10=150|
```

- FXCM Logon response

```
8=FIX.4.4|9=92|35=A|34=1|49=FXCM|50=U100D1|52=20120927-13:15:34.810|
↪56=fx1294946_client1|98=0|108=30|141=Y|10=187|
```

## 4.3 Marketdata

- Subscribe market price with 35=V MarketDataRequest. You will get MarketDataSnapshotFullRefresh(W) 35=W.
- The MarketDataSnapshotFullRefresh(W) message contains the updates to market data. It is obtained as a response to the MarketDataRequest(V) message. FIX connections are then subscription based for the market data; meaning, you must request it to receive it.
- The types of data you can receive, such as the Bid price or Offer price, are referred to as MDEntryTypes in FIX. FXCM supports the following MDEntryTypes in each message: Bid(0), Offer(1), High Price(7), and Low Price(8). Additional MDEntryTypes such as MDEntryDate, MDEntryTime, QuoteCondition, etc., are found only once within the first repeating group of the message.
- Also we don't have liquidity size information and depth information, only display BBO (best bid offer).
- If you want shrink price data which is 60% of current market price, please take a look at [here](#):

```
8=FIX.4.4|9=497|35=W|34=4|49=FXCM|50=U100D1|52=20180817-13:44:06.495|56=MD_d101968187_
↪client1|55=EUR/JPY|228=1|231=1|460=4|9001=3|9002=0.
↪01|9005=10|9011=0|9020=0|9080=1|9090=0|9091=0|9092=0|9093=0|9094=50000000|9095=1|9096=0|268=4|269=
↪085|271=0|272=20180817|273=13:44:06.000|336=FXCM|625=PSFX|276=A|282=PSFX_
↪DESK|299=FXCM-EURJPY-19288641|537=1|269=1|270=126.
↪093|271=0|272=20180817|273=13:44:06.000|336=FXCM|625=PSFX|276=A|282=PSFX_
↪DESK|299=FXCM-EURJPY-19288641|537=1|269=7|270=126.448|269=8|270=125.567|10=117|
```

If you want subscribe once and get update been pushed whenever there is update, you need set Subscription request type 263=1 which is snapshot + update. If 263=0, you will only get one snapshot.

You can also subscribe security in list, instead of just one security:

```
8=FIX.4.4|9=134|35=V|34=5|49=d101970033_client2|52=20180822-19:20:49.
↪137|56=FXCM|57=U100D1|262=4|263=1|264=3|265=0|146=1|55=EUR/
↪USD|267=2|269=0|269=1|10=189|8=FIX.4.4|9=808|35=V|34=5|49=MD_d101968187_
↪client1|52=20180817-13:44:05.
↪840|56=FXCM|57=U100D1|262=4|263=1|264=0|265=0|146=65|55=USOil|55=AUD/JPY|55=NZD/
↪CAD|55=EUR/CAD|55=USD/ZAR|55=AUS200|55=UKOil|55=EUR/NOK|55=NGAS|55=EUR/AUD|55=USD/
↪HKD|55=EUSTX50|55=GBP/CAD|55=USD/CAD|55=GER30|55=CAD/CHF|55=USD/TRY|55=EUR/
↪TRY|55=Copper|55=HKG33|55=USOilF2|55=GBP/AUD|55=NAS100|55=EUR/CHF|55=TRY/JPY|55=AUD/
↪NZD|55=USD/CHF|55=XAU/USD|55=FRA40|55=USOilF|55=AUD/USD|55=NZD/JPY|55=USD/
↪MXN|55=USDOLLAR|55=CHN50|55=ESP35|55=EUR/NZD|55=UKOilF|55=ZAR/JPY|55=GBP/CHF|55=NZD/
↪USD|55=USD/JPY|55=GBP/NZD|55=SPX500|55=CHF/JPY|55=UK100|55=EUR/USD|55=SOYF|55=GBP/
↪USD|55=EUR/JPY|55=AUD/CHF|55=EUR/GBP|55=XAG/USD|55=US30|55=GBP/JPY|55=NZD/
↪CHF|55=USD/NOK|55=CAD/JPY|55=AUD/CAD|55=Bund|55=USD/SEK|55=EUR/SEK|55=USD/
↪CNH|55=JPN225|55=UKOilF2|267=2|269=0|269=1|10=004|
```

## 4.4 Table

Send Collateral Inquiry 35=BB, you will get Collateral report 35=BA, which contains account information:

```
8=FIX.4.4|9=88|35=BB|34=4|49=d101970033_client2|52=20180817-20:17:27.
↪716|56=FXCM|57=U100D1|263=1|909=3|10=203|

8=FIX.4.4|9=361|35=BA|34=5|49=FXCM|50=U100D1|52=20180817-20:17:27.737|56=d101970033_
↪client2|1=01960313|53=1000|336=FXCM|625=U100D1|898=0|901=1000562.
↪37|908=4647057334|909=3|910=0|911=1|912=Y|921=1000562.37|922=1000562.
↪37|9038=0|9045=N|9046=0|9047=0|453=1|448=FXCMID|447=D|452=3|802=5|523=1960313|803=10|523=d101970033_
↪test138|803=22|523=32|803=26|523=Y|803=4000|10=033|
```

For Open positions, send Request for positions 35=AN with 724=0 (positions). You will get position report 35=AP for each open position. If you don't have open position, you will get "No open positions" in message 35=AO.

- Sample for open positions request:

```
8=FIX.4.4|9=149|35=AN|34=5|49=d101968168_client1|52=20151111-21:01:12.
↪396|56=FXCM|57=U100D1|1=01958448|60=20151111-21:01:12.
↪395|263=1|581=6|710=4|715=20151111|724=0|10=085|
```

- Sample for open position:

```
8=FIX.4.4|9=565|35=AP|34=8|49=FXCM|50=U100D1|52=20151111-20:19:59.929|56=d101968168_
↪client1|1=01958448|11=FIX.4.4:d101968168_client1->FXCM/U100D1-1437981786837-
↪10|15=EUR|37=207486895|55=EUR/USD|60=20150727-07:23:08|325=N|336=FXCM|526=fix_
↪example_
↪test|581=6|625=U100D1|710=4|715=20151111|721=3684204026|724=0|727=2|728=0|730=1.
↪10728|731=1|734=0|912=N|9000=1|9038=260|9040=-21.16|9041=80775478|9042=20150727-
↪07:23:08|9053=0.
↪8|453=1|448=FXCMID|447=D|452=3|802=4|523=32|803=26|523=d101968168|803=2|523=fix-
↪test112|803=22|523=1958448|803=10|702=1|703=TQ|704=10000|753=1|707=CASH|708=0|10=137|
```

For closed positions, please send positions request 35=AN ``with ``724=1 (Trades). You will receive position report in 35=AP.

- Sample for closed positions request:

```
8=FIX.4.4|9=177|35=AN|34=6|49=d101968168_client1|52=20151111-21:01:12.
↪400|56=FXCM|57=U100D1|1=01958448|60=20151111-21:01:12.
↪400|263=1|581=6|710=5|715=20151111|724=1|9012=20150311|9014=20151112|10=110|
```

- Sample for closed position:

```
8=FIX.4.4|9=702|35=AP|34=20|49=FXCM|50=U100D1|52=20151111-21:01:11.936|56=d101968168_
↪client1|1=01958448|11=FIX.4.4:d101968168_client1->FXCM/U100D1-1428599035518-
↪4|15=EUR|37=202027586|55=EUR/USD|60=20150519-03:30:43|325=N|336=FXCM|526=fix_
↪example_
↪test|581=6|625=U100D1|710=5|715=20151111|721=3533878441|724=1|727=13|728=0|730=1.
↪06572|731=1|734=0|912=Y|9000=1|9040=-6.08|9041=78911063|9042=20150409-
↪17:03:56|9043=1.12979|9044=20150519-03:30:43|9048=U100D1_
↪16679142D2EE08ABE053142B3C0A452A_05192015032653174913_QCV-127|9049=FXTS|9052=640.
↪7|9053=0.
↪8|9054=204437509|453=1|448=FXCMID|447=D|452=3|802=4|523=32|803=26|523=d101968168|803=2|523=fix-
↪test112|803=22|523=1958448|803=10|702=1|703=TQ|704=10000|753=1|707=CASH|708=0|10=042|
```

## 4.5 Order

Please set account number on tag 1, 1=00648329 when you place orders. Otherwise you will get error “No Account specified”. Place market order via 35=D. you will get execution report in 35=8:

- Open market position:

```
20160411-06:16:50.909 : 8=FIX.4.4 9=163 35=D 34=7 49=D101546502001_client1_
↪52=20160411-06:16:50.909 56=FXCM 57=U100D1 1=01537581 11=635959630109097564 38=10_
↪40=1 54=1 55=SPX500 59=1 60=20160411-06:16:50 10=054
```

- Sample execution report:

```
35=820160411-06:16:51.399: 8=FIX.4.4 9=478 35=8 34=15 49=FXCM 50=U100D1 52=20160411-
↪06:16:51.177 56=D101546502001_client1 1=01537581 6=2047.53 11=635959630109097564_
↪14=10 15=USD 17=821172034 31=2047.53 32=10 37=225909074 38=10 39=2 40=1 44=2047.53_
↪54=1 55=SPX500 58=Executed 59=1 60=20160411-06:16:51 99=0 150=F 151=0 211=0_
↪336=FXCM 625=U100D1 835=0 836=0 1094=0 9000=1010 9041=89603919 9050=OM 9051=F_
↪9061=0 453=1 448=FXCM ID 447=D 452=3 802=4523=1537581 803=10 523=d101546502001_
↪803=2 523=Halpert 803=22 523=32 803=26 10=088
```

## 4.6 FIX Sample Solutions and Projects

### 4.6.1 C++ FIX Example

Visual Studio 2010 Solution

Download QuickFIX FIX Engine C++ Solution

### 4.6.2 C++ QuickFIX Example VS2017 with x64

Visual Studio 2017 Solution

Download QuickFIX FIX Engine C++ x64 Solution



### 4.6.3 C# .NET FIX Example

Visual Studio 2010 Solution

[Download QuickFIX FIX Engine .Net Solution](#)

### 4.6.4 Java FIX Example

Netbeans Project

[Download QuickFIX/J FIX Engine Java Project](#)

### 4.6.5 C++ LAMM Trading Example

Visual Studio 2010 Solution

[Download QuickFIX FIX Engine for LAMM C++ Solution](#)

#### **Disclaimer**

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action."



## 5.1 TradingSessionStatusRequest (g)

After successful authentication, it is necessary to receive an update on the status of the market (open or closed), common system parameters, and a list of securities with their characteristics. The `TradingSessionStatus` message is returned as a response to the `TradingSessionStatusRequest` and it accomplishes these things.

## 5.2 TradingSessionStatus Message (h)

The `TradingSessionStatus` message is used to provide an update on the status of the market. Furthermore, this message contains useful system parameters as well as information about each trading security (embedded `SecurityList`).

`TradingSessionStatus` should be requested upon successful `Logon` and subscribed to. The contents of the `TradingSessionStatus` message, specifically the `SecurityList` and system parameters, should dictate how fields are set when sending messages to FXCM. For example, the instrument component within `SecurityList` indicates minimum order quantity 9095. Subsequent `NewOrderSingle (D)` messages should not violate this value when setting the `OrderQty` field (38).

- Requesting `TradingSessionStatus`

```
FIX44::TradingSessionStatusRequest request;  
request.setField(FIX::TradSesReqID(NextId()));  
request.setField(FIX::SubscriptionRequestType(FIX::SubscriptionRequestType_SNAPSHOT_  
↪PLUS_UPDATES));
```

- Reading System Parameters

Here we demonstrate how to extract the FXCM system properties from the `TradingSessionStatus` message. The code below will print out the name and value of each of the properties. The following custom fields are used:

```

9016 - FXCMNoParam
9017 - FXCMPParamName
9018 - FXCMPParamValue
int param_count = FIX::IntConvertor::convert(status.getField(9016));

cout << "TSS - FXCM System Parameters" << endl;
for(int i = 1; i <= param_count; i++)
{
    FIX::FieldMap map = status.getGroupRef(i, 9016);
    string param_name = map.getField(9017);
    string param_value = map.getField(9018);

    cout << param_name << " - " << param_value << endl;
}
FIX::Session::sendToTarget(request, session_id);

```

## 5.3 CollateralInquiry (BB)

CollateralInquiry is used to request the CollateralReport (BA) message from FXCM. This message contains important account related information such as the account number. With the exception of FIX sessions used solely for market data, you should include this message in your login sequence.

The login you use to connect will have access to one or more trading accounts. You will receive a CollateralReport for each of these accounts. When sending or modifying orders, you must set the Account(1) tag. This tag value must be set only to an account that you have actually received a CollateralReport for, otherwise you will see a rejection.

## 5.4 Requesting Market Data

The MarketDataSnapshotFullRefresh(W) message contains the updates to market data. It is obtained as a response to the MarketDataRequest(V) message. FIX connections are then subscription based for the market data; meaning, you must request it to receive it.

The types of data you can receive, such as the Bid price or Offer price, are referred to as MDEntryTypes in FIX. FXCM supports the following MDEntryTypes in each message: Bid(0), Offer(1), High Price(7), and Low Price(8). Additional MDEntryTypes such as MDEntryDate, MDEntryTime, QuoteCondition, etc., are found only once within the first repeating group of the message.

- Sending MarketDataRequest(V) Message

```

FIX44::MarketDataRequest mdr;

mdr.set(FIX::MDReqID(NextId()));
mdr.set(FIX::SubscriptionRequestType(FIX::SubscriptionRequestType_SNAPSHOT_PLUS_
↳UPDATES));
mdr.set(FIX::MarketDepth(0));
mdr.set(FIX::NoMDEntryTypes(2));

FIX44::MarketDataRequest::NoMDEntryTypes types_group;
types_group.set(FIX::MDEntryType(FIX::MDEntryType_BID));
mdr.addGroup(types_group);
types_group.set(FIX::MDEntryType(FIX::MDEntryType_OFFER));
mdr.addGroup(types_group);

```

(continues on next page)

(continued from previous page)

```

int no_sym = FIX::IntConvertor::convert(security_list.
    ↳getField(FIX::FIELD::NoRelatedSym));
for(int i = 1; i <= no_sym; i++)
{
    FIX44::SecurityList::NoRelatedSym sym_group;
    mdr.addGroup(security_list.getGroup(i, sym_group));
}

FIX::Session::sendToTarget(mdr, session_id);

```

## 5.5 Getting Positions

Open and closed positions are retrieved through the `PositionReport (AP)` message. Unlike the `ExecutionReport (8)` message which contains information relating to orders, the `PositionReport` is not automatically sent to your FIX client. You can make individual requests for `PositionReport`, or you can subscribe to updates on this message. Sending a `RequestForPositions (AN)` message with `SubscriptionRequestType (263)` set to 1 (`SnapshotAndUpdates`) will subscribe to updates.

### 5.5.1 Open Position vs. Closed Position

`PosReqType (724)` is used to determine if a received `PositionReport` represents an open position or closed position. A value of 0 indicates an open position while a value of 1 indicates a closed position.

#### 5.5.1.1 Open Positions

When a `PositionReport` representing an open position is sent to you, it will contain the price at which the position was opened. This can be seen using `SettlPrice (730)`. The close price and the P/L of the position are not present given that the position is open. Close price and P/L are real-time calculated values and are not contain in any `PositionReport` where `PosReqType (724) = 0` (Open Position).

#### 5.5.1.2 Closed Positions

`PositionReport` messages representing closed positions will include the price at which the position was closed, as well as other useful information such as P/L. The following additional tags are present:

Tag	Description
FXCMPosClosePNL (9052)	Gross P/L of the position; e.g., \$24.17, or €43.72
FXCMPosInterest (9040)	Rollover interest applied to the position
FXCMPosCommission (9053)	Commission applied to the position
FXCMCloseSettlPrice (9043)	Close price of the position

#### 5.5.1.3 Position Margin

The margin applied to each individual position can be obtained from a `PositionReport` representing an open position. `FXCMUsedMargin (9038)` will contain this margin value. Note that the total margin required for an account can be obtained by this same tag from the `CollateralReport (BA)` message.

## 5.6 Overview of Basic Order and Time-In-Force Types

### 5.6.1 Time-In-Force (TIF) Types

The `Time-In-Force (59)` Tag is used to indicate how long an order should remain active before it is either executed by the broker or cancelled by the client. Below are the four TIF values with descriptions.

#### 5.6.1.1 Good Til Cancel (GTC)

Orders with this TIF value remain open and active until fully executed or cancelled. This means that the order remains active until the entire order amount is executed.

##### When to use GTC

Use GTC when your order must remain active until it can be filled Use GTC when your entire order must get filled

#### 5.6.1.2 Day

Orders with this TIF value will remain open and active until fully executed, cancelled by the client, or when the trading day ends. Like Good Til Cancel (GTC), this means the order will remain active until the entire order amount is executed, unless the order is cancelled or the trading day ends.

##### When to use Day

Use Day when your original intention for the order becomes obsolete with time.

#### 5.6.1.3 Immediate or Cancel (IOC)

Orders with this TIF value will immediately attempt to execute as much of your order as possible and cancel any remaining amount if necessary. As a result, this TIF value will allow partial fills.

##### When to use IOC

Use IOC when you expect execution to take place immediately Use IOC when it is acceptable if your entire order does not get filled

#### 5.6.1.4 Fill or Kill (FOK)

Orders with this TIF value will attempt to execute the entire order amount immediately. If the entire order amount cannot be executed, the order is cancelled.

##### When to use FOK

Use FOK when you expect execution to take place immediately Use FOK when your entire order must get filled

### 5.6.2 Order Types

#### 5.6.2.1 Market

A market order is an order to buy or sell immediately at the next available price. This means that the order is not guaranteed to fill at any specific price.

##### When to Use Market

Use market when your order being filled is more important than the price it is filled at.

#### Supported TIF Values

GTC, DAY, IOC, **and** FOK

#### 5.6.2.2 Market Range (Stop-Limit)

Market range is a market order that comes with a limitation on the price at which the order can be filled. In other words, it is a market order with a protection against slippage (price deviation).

In FIX terms, you can convert a market order to a market range order by setting the `OrdType (40)` to 4 (`Stop-Limit`) and by setting the `StopPx (99)` Tag. The `StopPx` tag value should be set to the worst price you would accept being filled.

#### Example

Assume you want to Buy EUR/USD now while it is trading at 1.4531 but you do not want to get filled at a price higher than 1.4535. In this case you would set the `StopPx (99)` tag value to 1.4535. If your order cannot be filled at 1.4535 or below, it will be cancelled.

#### When to use Market Range

Use market range when you are concerned about slippage Use market range when it is acceptable that your order may be cancelled

#### Supported TIF Values

IOC **and** FOK

#### 5.6.2.3 Limit

A limit order is an order to buy or sell only at a specific price (or better). In other words, the order can only be filled at the limit price or for some better price.

#### When to use Limit

Use limit when you must guarantee the price at which an order is filled

#### Supported TIF Values

GTC, Day, IOC, **and** FOK

#### Common Applications

The limit order can be used to achieve multiple objectives when combined with different TIF values. The two common application are:

#### 5.6.2.4 GTC/Day Limit Order

Recall that both GTC and DAY remain active until the entire order is filled, until cancelled, or the trading day ends (for DAY orders). When you combine the limit order with these TIF values, you have an order that will remain active until the entire amount is filled at your limit price or better. This type of order is often used to close an existing position and ensure the position is closed at a specific rate.

### 5.6.2.5 IOC/FOK Limit Order

Recall that with IOC and FOK, your order will immediately attempt execution. In the case of IOC, part of the order will be filled if possible. In the case of FOK, the entire order must be filled. When you combine IOC/FOK with the limit order, you have an order which will attempt execution immediately but will fill only at your limit price or better. This order type is commonly used to guarantee the price at which a new order is filled while also controlling how much can be filled; IOC would allow partial fills while FOK would not.

### 5.6.2.6 Stop

A stop is an order to buy or sell some amount when the current market price reaches your stop price. In other words, a stop order is a market order which is waiting to be active until the market price reaches a certain level (your stop price). Given that a stop is effectively a type of market order, it does not guarantee any specific fill price.

#### When to use Stop

Use stop when your order being filled is more important than the price it is filled at

#### Supported TIF Values

GTC and DAY

### 5.6.2.7 Trailing Stop Peg Order

**Stop/Limit Order**

Ticket: 47786431

☒ Stop:  
 Rate: 1.12282 >1.11891 ☐ In pips  
 Trailing: None 0

☒ Limit:  
 Rate: 1.11708 <1.11891 ☐ In pips

Profit/Loss  
 Stop Loss: -39.40 @ 1.12282  
 Limit Profit: 18.00 @ 1.11708

OK Cancel Chart>>

#### How to set stop peg orders

First You need to set ELS order, then in stop order please set stop price on tag 99 and fluctuate point at tag 9061.

9061=1 == dynamic 9061 = from 2 to 9 is invalid (you will get error "Trailing step did not pass validation.") 9061=10+ == fixed

#### How does peg orders work



ELS order with main order sell 1K USD/JPY market FOK + stop order at 99=104.504 with fluctuate point 9061=10 and limit order. Please be aware that this ELS has three orders bind together. Each order should has its own CLOrdID In this case, the stop order is 11 = 1475761911686.

```
11=1475761911686|67=1|583=2|1=1206026806|55=USD/JPY|54=1|60=20161006-13:51:51.
↪682|38=1000|40=3|99=104.504|9061=10|
```

Main order been filled immediately at market. At the same time market rate of ask been set to tag 31 = 104.004 (this is not trigger price but you can think this is trigger reference, you can get this value in market price message 35=W) and stop price at 104.504 at tag 99.

```
11=1475761911686|14=0|15=USD|17=111400333|31=104.
↪004|32=0|37=60513656|38=1000|39=0|40=P|44=104.504|54=1|55=USD/JPY|59=1
```

When market moved to 103.904 which is 10 points from this reference 104.004, both market price and stop price been moved 10 point. And becomes as tag 31 = 103.904 and tag 44 = 104.404 You will see the stop price change on TSII GUI also. Then it wait the market to touch 10 point off 103.904 which is 103.804.

```
11=1475761911686|14=0|15=USD|17=111402690|31=103.
↪904|32=0|37=60513656|38=1000|39=0|40=P|44=104.404|54=1|55=USD/JPY|59=1|
```



## ORDER ACTIVITY

Order #	Type	TIF	Date of Operation	Status	Ticket #	Volume	Price	Value Date	Expire Date	Trail Step	Market Rate	Price Stream	Operation by	Operator's Address
60513656	S(t)	GTC	10/6/16 2:03 PM	Waiting	47740717	1K	104.404			10	103.904		U100R4_EXOTIC	10.60.82.46:2678789
60513656	S(t)	GTC	10/6/16 1:51 PM	Waiting	47740717	1K	104.504			10	104.004		1206026806	10.60.82.46:2677878
60513656	S(t)	GTC	10/6/16 1:51 PM	Waiting	47740717	1K	104.504			10	104.004		1206026806	cust110.nydc.fxcorp.prv

```
DEBUG (2016-10-06 09:51:51,696) [QF/J Session dispatcher: FIX.4.4:FXCM/RAPID->
↪1206026806_client1] (app) - <<< app message from counterparty: 8=FIX.4.
↪4|9=430|35=E|34=5|49=1206026806_client1|52=20161006-13:51:51.
↪682|56=FXCM|57=RAPID|66=1475761911669|68=3|1385=101|73=3|
11=1475761911685|67=0|583=1|1=1206026806|55=USD/JPY|54=2|60=20161006-13:51:51.
↪682|38=1000|40=1|59=4|
```

(continues on next page)

(continued from previous page)

```

11=1475761911686|67=1|583=2|1=1206026806|55=USD/JPY|54=1|60=20161006-13:51:51.
↪682|38=1000|40=3|99=104.504|9061=10|
11=1475761911687|67=2|583=2|1=1206026806|55=USD/JPY|54=1|60=20161006-13:51:51.
↪682|38=1000|40=2|44=103.504|10=087|

```

```

DEBUG (2016-10-06 09:51:51,702) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=518|35=8|34=14|49=FXCM|50=RAPID|52=20161006-
↪13:51:51.702|56=1206026806_client1|1=1206026806|6=104.
↪504|11=1475761911686|14=0|15=USD|17=111400333|31=104.
↪004|32=0|37=60513656|38=1000|39=0|40=P|44=104.504|54=1|55=USD/JPY|59=1|60=20161006-
↪13:51:51|66=1475761911669|99=0|150=0|151=1000|198=60513655|211=104.
↪504|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=2|9041=47740717|9050=ST|9051=W|9061=10|907
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -
↪Test|803=22|523=32|803=26|10=105|

```

```

DEBUG (2016-10-06 09:51:51,826) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=505|35=8|34=19|49=FXCM|50=RAPID|52=20161006-
↪13:51:51.826|56=1206026806_client1|1=1206026806|6=104.
↪504|11=1475761911686|14=0|15=USD|17=111400508|31=104.
↪004|32=0|37=60513656|38=1000|39=0|40=P|44=104.504|54=1|55=USD/JPY|59=1|60=20161006-
↪13:51:51|66=1475761911669|99=0|150=0|151=1000|211=104.
↪504|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=2|9041=47740717|9050=ST|9051=W|9061=10|907
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -
↪Test|803=22|523=32|803=26|10=246|

```

```

DEBUG (2016-10-06 10:03:26,724) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=505|35=8|34=33|49=FXCM|50=RAPID|52=20161006-
↪14:03:26.724|56=1206026806_client1|1=1206026806|6=104.
↪404|11=1475761911686|14=0|15=USD|17=111402690|31=103.
↪904|32=0|37=60513656|38=1000|39=0|40=P|44=104.404|54=1|55=USD/JPY|59=1|60=20161006-
↪14:03:26|66=1475761911669|99=0|150=0|151=1000|211=104.
↪404|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=2|9041=47740717|9050=ST|9051=W|9061=10|907
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -
↪Test|803=22|523=32|803=26|10=248|

```

### Dynamic peg example: “9061=1”

```

8=FIX.4.4|9=515|35=E|34=6|49=1206026806_client1|52=20161007-14:53:41.
↪963|56=FXCM|57=RAPID|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|68=3|1385=101|73=3|11=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪1475852021962-6|67=0|583=1|1=1206026806|55=EUR/USD|54=2|38=10000|40=1|11=FIX.4.
↪4:1206026806_client1->FXCM/RAPID-1475852021962-7|67=1|583=2|1=1206026806|55=EUR/
↪USD|54=1|38=10000|40=3|99=1.12401|9061=1|11=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪1475852021963-8|67=2|583=2|1=1206026806|55=EUR/USD|54=1|38=10000|40=2|44=1.
↪11725|10=228|

```

### Peg been triggered, price in tag “44” is in dynamic

```

DEBUG (2016-10-07 10:53:41,821) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=600|35=8|34=69|49=FXCM|50=RAPID|52=20161007-
↪14:53:41.821|56=1206026806_client1|1=1206026806|6=1.12401|11=FIX.4.4:1206026806_
↪client1->FXCM/RAPID-1475852021962-7|14=0|15=EUR|17=111563733|31=1.
↪11922|32=0|37=60576267|38=10000|39=0|40=P|44=1.12401|54=1|55=EUR/
↪USD|59=1|60=20161007-14:53:41|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|99=0|150=0|151=10000|198=60576266|211=1.
↪12401|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=1|9041=47788992|9050=ST|9051=W|9061=1|907
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -
↪Test|803=22|523=32|803=26|10=244|

```

(continues on next page)

(continued from previous page)

```

DEBUG (2016-10-07 10:53:41,959) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=587|35=8|34=75|49=FXCM|50=RAPID|52=20161007-
↪14:53:41.959|56=1206026806_client1|1=1206026806|6=1.12401|11=FIX.4.4:1206026806_
↪client1->FXCM/RAPID-1475852021962-7|14=0|15=EUR|17=111563912|31=1.
↪11922|32=0|37=60576267|38=10000|39=0|40=P|44=1.12401|54=1|55=EUR/
↪USD|59=1|60=20161007-14:53:41|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|99=0|150=0|151=10000|211=1.
↪12401|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=1|9041=47788992|9050=ST|9051=W|9061=1|9079=
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -_
↪Test|803=22|523=32|803=26|10=132|

DEBUG (2016-10-07 10:53:42,060) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=581|35=8|34=77|49=FXCM|50=RAPID|52=20161007-
↪14:53:42.060|56=1206026806_client1|1=1206026806|6=1.124|11=FIX.4.4:1206026806_
↪client1->FXCM/RAPID-1475852021962-7|14=0|15=EUR|17=111563914|31=1.
↪11921|32=0|37=60576267|38=10000|39=0|40=P|44=1.124|54=1|55=EUR/USD|59=1|60=20161007-
↪14:53:42|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|99=0|150=0|151=10000|211=1.
↪124|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=1|9041=47788992|9050=ST|9051=W|9061=1|9079=
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -_
↪Test|803=22|523=32|803=26|10=079|

DEBUG (2016-10-07 10:53:42,571) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=587|35=8|34=78|49=FXCM|50=RAPID|52=20161007-
↪14:53:42.571|56=1206026806_client1|1=1206026806|6=1.12396|11=FIX.4.4:1206026806_
↪client1->FXCM/RAPID-1475852021962-7|14=0|15=EUR|17=111563915|31=1.
↪11917|32=0|37=60576267|38=10000|39=0|40=P|44=1.12396|54=1|55=EUR/
↪USD|59=1|60=20161007-14:53:42|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|99=0|150=0|151=10000|211=1.
↪12396|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=1|9041=47788992|9050=ST|9051=W|9061=1|9079=
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -_
↪Test|803=22|523=32|803=26|10=173|

DEBUG (2016-10-07 10:53:43,131) [0:Bus:1206026806_client1FXCMRAPID] (app) - >>> app_
↪message to counterparty: 8=FIX.4.4|9=587|35=8|34=79|49=FXCM|50=RAPID|52=20161007-
↪14:53:43.131|56=1206026806_client1|1=1206026806|6=1.12393|11=FIX.4.4:1206026806_
↪client1->FXCM/RAPID-1475852021962-7|14=0|15=EUR|17=111563916|31=1.
↪11914|32=0|37=60576267|38=10000|39=0|40=P|44=1.12393|54=1|55=EUR/
↪USD|59=1|60=20161007-14:53:43|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758520217125|99=0|150=0|151=10000|211=1.
↪12393|336=FXCM|625=RAPID|835=0|836=0|1094=0|1385=101|9000=1|9041=47788992|9050=ST|9051=W|9061=1|9079=
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -_
↪Test|803=22|523=32|803=26|10=157|

```

**Invalid peg example**

```

8=FIX.4.4|9=515|35=E|34=6|49=1206026806_client1|52=20161007-16:06:30.
↪401|56=FXCM|57=RAPID|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758563901965|68=3|1385=101|73=3|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475856390400-
↪6|67=0|583=1|1=1206026806|55=EUR/USD|54=2|38=10000|40=1|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475856390401-
↪7|67=1|583=2|1=1206026806|55=EUR/USD|54=1|38=10000|40=3|99=1.12019|9061=2|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475856390401-
↪8|67=2|583=2|1=1206026806|55=EUR/USD|54=1|38=10000|40=2|44=1.11344|10=236|

```

(continues on next page)

(continued from previous page)

```

8=FIX.4.4|9=557|35=8|34=7|49=FXCM|50=RAPID|52=20161007-16:06:30.470|56=1206026806_
↪client1|1=1206026806|6=1.11524|11=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪1475856390400-6|14=0|15=EUR|17=111586686|31=1.
↪11524|32=0|37=60582259|38=10000|39=0|40=1|44=1.11524|54=2|55=EUR/
↪USD|59=1|60=20161007-16:06:30|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758563901965|99=0|150=0|151=10000|211=0|336=FXCM|625=RAPID|835=0|836=0|1094=0|9000=1|9041=477935|
↪ID|447=D|452=3|802=4|523=6026806|803=10|523=1206026806|803=2|523=API -
↪Test|803=22|523=32|803=26|10=156|

8=FIX.4.4|9=646|35=8|34=8|49=FXCM|50=RAPID|52=20161007-16:06:30.471|56=1206026806_
↪client1|1=1206026806|6=1.12019|11=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪1475856390401-7|14=0|15=EUR|17=0|31=0|32=0|37=NONE|38=10000|39=8|40=3|44=1.
↪12019|54=1|55=EUR/USD|58=19915;DAS 19915: ZDas Exception ORA-20115: Traling step
↪did not pass validation.|59=1|60=20161007-16:06:30|66=FIX.4.4:1206026806_client1->
↪FXCM/RAPID-14758563901965|99=1.
↪12019|103=99|150=8|151=0|211=0|336=FXCM|625=RAPID|835=0|836=0|1094=0|9000=1|9025=0|9029=19915;
↪DAS 19915: ZDas Exception ORA-20115: Traling step did not pass validation.
↪|9051=R|9061=0|453=1|448=FXCM ID|447=D|452=3|802=1|523=6026806|803=10|10=153|

```

### Fixed peg example

```

8=FIX.4.4|9=516|35=E|34=6|49=1206026806_client1|52=20161007-13:44:16.
↪267|56=FXCM|57=RAPID|66=FIX.4.4:1206026806_client1->FXCM/RAPID-
↪14758478560675|68=3|1385=101|73=3|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475847856266-
↪6|67=0|583=1|1=1206026806|55=EUR/USD|54=2|38=10000|40=1|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475847856267-
↪7|67=1|583=2|1=1206026806|55=EUR/USD|54=1|38=10000|40=3|99=1.12382|9061=10|
11=FIX.4.4:1206026806_client1->FXCM/RAPID-1475847856267-
↪8|67=2|583=2|1=1206026806|55=EUR/USD|54=1|38=10000|40=2|44=1.11708|10=103|

```

## 5.7 Handling of Partial Fills

### 5.7.1 Order Quantity Fields

There are three fields which can be used to determine the quantity filled or rejected by FXCM. These fields are:

- LastQty (32) – the quantity filled on the last successful attempt to fill the order
- CumQty (14) – the total quantity filled
- LeavesQty (151) – the remaining quantity to be filled
- Importance of OrdStatus (39)

It is important to consider OrdStatus (39) when using the quantity fields above. As FXCM is attempting to execute an order, the values of OrdStatus will progress from an initial value of New (0) to some final state. There are three possible final values for OrdStatus:

- OrdStatus = Filled (2)
- OrdStatus = Rejected (8)
- OrdStatus = Cancelled (4)

When you receive an ExecutionReport 8 with OrdStatus set to one of these final values, you can inspect the CumQty (14) field to determine the total amount executed. If OrdStatus = Filled (2), the entire order was filled and CumQty will equal the original OrdQty value. If OrdStatus = Rejected (8), the order was partially filled and CumQty will be some value less than the original OrdQty.

## 5.7.2 Example Partial Fill

The following ExecutionReport messages serve as an example of a partially filled order. The original OrderQty(38) was 1,000,000. In this example only 600,000 of the order was filled. The most important line here is the last, where we can see a final OrdStatus value (Rejected in this case). When this last ExecutionReport is received, we can inspect CumQty (14) to see that 600,000 was filled.

```
OrderQty(38) = 1,000,000; OrdStatus(39) = New
6=85.558 14=0 17=59342024 31=85.558 32=0 37=31654622 38=1000000 39=0 40=4 44=85.55
↪854=2 59=3 99=0 150=0 151=1000000 211=0835=0 836=0 1094=0 9000=17 9041=13151786
↪9050=OR 9051=P 9061=0

OrderQty(38) = 1,000,000; OrdStatus(39) = Stopped
6=85.558 14=0 17=59342025 31=85.558 32=0 37=31654622 38=1000000 39=7 40=4 44=85.55
↪854=2 59=3 99=0 150=7 151=1000000 211=0835=0 836=0 1094=0 9000=17 9041=13151786
↪9050=OR 9051=U 9061=0

OrderQty(38) = 1,000,000; OrdStatus(39) = Stopped
6=85.488 14=0 17=59342047 31=85.488 32=0 37=31654622 38=1000000 39=7 40=4 44=85.48
↪854=2 9=3 99=0 150=7 151=1000000 211=0835=0 836=0 1094=0 9000=17 9041=13151786
↪9050=OR 9051=U 9061=0

OrderQty(38) = 1,000,000; OrdStatus(39) = Partially Filled; LastQty(32) = 600,000;
↪CumQty(14) = 600,000
6=85.488 14=600000 17=59342048 31=85.488 32=600000 37=31654622 38=1000000 39=1 40=4
↪44=85.488 54=259=399=0 150=F 151=400000 211=0 835=0 836=0 1094=0 9000=17
↪9041=13151888 9050=OR 9051=U 9061=0

OrderQty(38) = 1,000,000; OrdStatus(39) = Rejected; CumQty(14) = 600,000
6=85.488 14=600000 17=59342049 31=85.488 32=0 37=31654622 38=1000000 39=8 40=4 44=85.
↪488 54=258=Rejected 59=399=0 150=8 151=0 211=0 835=0 836=0 1094=0 9000=17
↪9041=13151888 9050=OR 9051=R 9061=0
```

## 5.7.3 Closing A Position

How you close a position depends upon the position maintenance type of the account. Some accounts support hedging while others do not. Hedging is the ability to have two positions in the same symbol but of a different side; for example, holding both Buy EUR/USD and Sell EUR/USD positions at the same time.

### Accounts with Hedging

Accounts that support hedging allow you to close individual positions, regardless of when they were opened relative to other positions. Clearly with these accounts, Buy and Sell orders do not offset themselves but instead form a hedge. Consequently, you must close these positions with a NewOrderSingle message that specifies the TicketID to close.

## 5.7.4 Sending A Closing Order

NewOrderSingle (D) can be used to close a specific position simply by setting the FXCMPoSID (9041) field. This converts a basic market order into a closing order.

- Closing Order in Code

```
FIX44::NewOrderSingle order;

order.setField(ClOrdID(NextClOrdID()));
order.setField(Account(account_ID));
order.setField(Symbol("EUR/USD"));
order.setField(Side(Side_BUY));
order.setField(TransactTime());
order.setField(OrderQty(10000));
order.setField(OrdType(OrdType_MARKET));
order.setField(FXCM_POS_ID/*9041*/, "84736256");

Session::sendToTarget(order, session_ID);
```

### Accounts without Hedging

For accounts without hedging, orders of the opposite Side cancel each other out; e.g., sending a NewOrderSingle with a Side of Buy will net against any existing Sell positions. This netting is done in First-In, First-Out FIFO order. As a result, a basic market order will suffice to close any open position.

## 5.8 Getting Account Position Maintenance

The position maintenance type of each account can be retrieved from the Parties component of CollateralReport (BA). The NoPartySubIDs group contains a custom PartySubIDType for position maintenance. This specific PartySubIDType tag is set to a value of 4000. PartySubID can be checked for the value of position maintenance. Y = "Hedging Enabled," N = "No Hedging," and 0 = "Netting." Anything other than Y implies hedging is disabled and we will not use closing orders.

### 5.8.1 Getting Position Maintenance in Code

```
int number_subID = IntConvertor::convert(group.getField(FIELD::NoPartySubIDs));
for(int u = 1; u <= number_subID; u++){
    FIX44::CollateralReport::NoPartyIDs::NoPartySubIDs sub_group;
    group.getGroup(u, sub_group);

    string sub_type = sub_group.getField(FIELD::PartySubIDType);
    string sub_value = sub_group.getField(FIELD::PartySubID);
    if(sub_type == "4000"){
        // Check sub_value for position maintenance
        // Y = Hedging
        // N = No Hedging
        // 0 = Netting
    }
}
```

## 5.9 When To Reset MsgSeqNum

### 5.9.1 Reset On Logon

MsgSeqNum should be reset upon each Logon. This means that every Logon message should include tags MsgSeqNum (34) set to “1” and ResetSeqNumFlag (141) set to “Yes.” It is necessary to reset upon each Logon due to the fact that connections to FXCM are load balanced against a cluster of servers. This promotes a stable trading environment for users, but it also means you should reset upon each Logon.

Example Logon Message:

```
8 (BeginString)=FIX.4.4
9 (BodyLength)=114
35 (MsgType)=A
34 (MsgSeqNum)=1
49 (SenderCompID)=sender_client1
52 (SendingTime)=20120927-13:15:34.754
56 (TargetCompID)=FXCM
57 (TargetSubID)=U100D1
553 (Username)=some_user
554 (Password)=some_password
98 (EncryptMethod)=0
108 (HeartBtInt)=30
141 (ResetSeqNumFlag)=Y
10 (Checksum)=146
```

## 5.10 Account Equity

FIX API does not have a field which represents account equity. Equity is a real-time value that is dependent upon a floating price. If your application needs immediate access to equity in real-time, you would have to calculate it using market data. However, the CollateralReport (BA) does provide an equity value that corresponds with a specific time in the trading day.

### 5.11 StartCash(921)

The StartCash (921) field from CollateralReport is the equity value of the account at 5:00pm EST (New York). This can be used as a snapshot of what the equity was at that time. This value will include the account balance and any profit or loss on open trades.

## 5.12 Steps To Retrieve Short Version Of Market Price

FXCM give client an opportunity to retrieve market price for just Bid/Ask, please follow instructions at [here](#).

### 5.13 EMF

**Note:** Important detail to note about order execution with FXCM is the difference between order fill notification and order finished notification. As an order is filled by a liquidity provider, client will be sent a fill confirmation in the form

of an execution report that includes 35=8|39=7|150=F or, in case of a partial fill, 35=8|39=1|150=F. This confirmation is sent as soon as the LP confirms the trade. After the order is completed and every database operation associated with it is committed, the client will be sent an execution report of order being done. This execution report includes 35=8|39=2|150=F. Alternatively, if the order was filled only partially before being canceled, the final confirmation will include 35=8|39=4|150=4. You can find the remaining quantity that was not filled in tag 151. It is important to note, that the final execution report can be sent much later. When looking for fill confirmations, clients can take advantage of faster notifications than before implementing EMF. Even if clients are not taking advantage of the EMF execution, they will always be notified of the orders being filled. The only difference would be the delivery delay.

---

## 5.14 Execution Disclaimer

FXCM aggregates bid and ask prices from a pool of liquidity providers and is the final counterparty when trading forex on FXCM's dealing desk and No Dealing Desk (NDD) execution models. With NDD, FXCM's platforms display the best-available direct bid and ask prices from the liquidity providers. In addition to the spread, the trading cost with NDD is a fixed lot-based commission at the open and close of the trade. While generally NDD accounts offer spreads with no markups, in some circumstances, FXCM may add a markup to NDD spreads. This may occur due to, but not limited to, account type, such as accounts opened through a referring agent. With dealing desk execution, FXCM can act as the dealer on any or all currency pairs. Backup liquidity providers fill in when FXCM does not act as the dealer. FXCM's dealing desk has fewer liquidity providers than NDD. There are many other factors to consider when choosing an execution model (such as conflict of interest, trading style or strategy). See Execution Risks. Note: Contractual relationships with liquidity providers are consolidated through the FXCM Group, which, in turn, provides technology and pricing to the group affiliate entities.

### Disclaimer

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action."



## Frequently Asked Questions

### 6.1 Error Messages received

#### ORA-20103: Session expired

```
SEVERE: onMessageRecieved::Forcereconnect from server. Update session state::19915;
↪DAS 19915: ZDas Exception ORA-20103: Session expired.
```

Unordered List Item Error code 20103 session expired means your connection has been lost. This error message could be displayed due to a number of reasons, including network instability, a system issue or a client side program crash. If the problem is a system issue, please try to reboot.

#### ORA-20143: Order price too far from market price

```
Done! request.RequestID: U10D1_0F742A280DE8276EE053182B3C0A1526_02192015163720178345_
↪XIX0-2 offer.OfferID:1 AccountID: 831293 iAmount: 10000 dRate: 1.13953 dRateLimit:
↪1.14153 dRateStop: 1.13803 BuySell: B OrderType: LE 19915;DAS 19915: ZDas Exception
↪ORA-20143: Order price too far from market price :1.14219 vs 1.14162
```

Unordered List Item This error message is generated when the Buy Limit price is above the Bid price. For example if the Bid price was 1.13919, and your Buy limit 1.13953. If you want to place a Buy Limit above the Bid price you can do so using an OpenLimit order, which is available in API ver. 1.3.2. The fill price will be the limit price or better.

#### ORA-20112: Limit price did not pass validation

```
19915;DAS 19915: ZDas Exception ORA-20112: Limit price did not pass validation:
↪A:308386 OF:22 SB:B
```

Unordered List Item This error message is generated when the Limit price does not correspond to the ask price for the order type required. If the Time in Force is IOC or FOK then the Buy limit price should be  $\geq$  Ask price. For GTC or GTD the Buy Limit should be Ask price.

#### ORA-20113: Insufficient margin in session

```
19915;DAS 19915: ZDas Exception ORA-20113: Insufficient margin in session
```

Unordered List Item This error message is generated when you don't have enough margin.

#### **ORA-20102: Access Violation**

```
19915;DAS 19915: ZDas Exception ORA-20102: Access Violation: U10D1_
→0C73B32A0A4299C5E053182B3C0ADFC8_01122015125318715906_UGQ1 phase:4
```

Unordered List Item This error message is generated when a trade account is missing from the dealer account.

#### **ORA-20105: Order price did not pass validation**

```
8=FIX.4.4|9=179|35=D|1=2620231783|11=FXSTAT_
→32951421147085|15=EUR|38=5|40=3|54=2|55=ESP35|59=1|60=20150113-
→11:04:45|99=9864|386=1|336=FXCM|625=EUREAL|516=0|526=OT_-2_1421147086_ESP35_ESP35_0_
→0_|10=080
```

Unordered List Item The rejected orders error message is generated when the stop price is within 12 points of ask price. The reason for rejected orders is because the Minimum Stop Distance for ESP35 is 12 points. For example, if the Ask price was 9911 and your Stop price 99=9917, you would receive this error message. In this example, the stop price should be at least  $9911 + 12 = 9923$ .

#### **ORA-20008: Failed to create order, primary validation**

This error message is generated when Range prices are below the Ask price. For example if orders were placed on news events, and the spreads got wider, e.g. If Buy range IOC 55=CAD/JPY 44=92.55100 99=95.55100 and Ask price was 95.612

```
8=FIX.4.4 9=190 35=D 34=19 49=2620237129_client1 52=20150304-15:00:00 56=FXCM_
→57=EUREAL 1=2620237129 11=28020200337fa23 38=1000000 40=4 44=92.55100 54=1 55=CAD/
→JPY 59=3 60=20150304-14:57:19.363 99=95.55100 10=252

DEBUG (2015-03-04 10:00:00,171) [0:Bus:2620237129_client1FXCMEUREAL] (app) - »> app_
→message to counterparty: 8=FIX.4.4 9=558 35=8 34=18308 49=FXCM 50=EUREAL_
→52=20150304-15:00:00.171 56=2620237129_client1 1=2620237129 6=0 11=28020200337fa23_
→14=0 15=CAD 17=0 31=0 32=0 37=NONE 38=1000000 39=8 40=4 44=0 54=1 55=CAD/JPY_
→58=19915;DAS 19915: ZDas Exception ORA-20008: Failed to create order, primary_
→validation
```

#### **tag specified out of required order**

This error message is generated when the tag order does not pass our system check. You can avoid this by setting ValidateFieldsOutOfOrder=N in your config file.

#### **How can I tell what account type I have?**

Checking on Trading station: To check the type of account you have, you can login to Trading station and look in the tab "Accounts". Scroll to the end and find column Type. Y = Hedging is allowed; N = Hedging is not allowed, O = Netting only, D = Day netting, F = FIFO

FIX: Using tag 453 and 803 (PartySubIDType) Y = Hedging, N = No Hedging, O = Netting. This is located on page 34 of the documentation.

## **6.2 Limit order Day order vs Limit IOC/FOK**

A Limit Order is an order to buy or sell a predetermined amount at a specified price. This order will be filled only when the market price equals the specified limit price or better. Limit orders also allow a trader to limit the length

of time an order can be outstanding before being canceled with the following time in force values GTC, DAY, GTD, IOC, FOK.

The Limit price for order with TIF GTC or DAY (for future execution) should be market or better than current market price, but for orders with TIF IOC or FOK (immediate execution) the Limit price should be market or worse. This is because IOC/FOK orders will be sent immediately for execution, without waiting for the market price to reach the Limit price.

If the current Bid is 100, and you place a Sell Limit IOC/FOK at 102, this order will be rejected because the Limit price has to be market or worse for IOC/FOK, (it would be like asking to get order filled immediately at 102 or better).

If your intention was to control the slippage, you can use a Limit OIC or FOK. The Limit price should be the Bid price or below. So in this example, if Limit=98, the order will be filled at 98 or better, but because it's immediate execution, if for any reason the order can't be filled in few attempts, the order will be canceled. Can you fix the paragraph spacing below?

## 6.3 What is the Base Unit Size (also the minimum trade size) for all FX instruments?

tag 53 in collateral report(35=BA) For each CFD - tag 228 in Trading Session Status 35=h

## 6.4 How can I get Positions side and quantity from open position report?

First you need to send position report request 35=AN with 724=0 (open position), 724=1 is for closed positions. If you have open position you will get positon report 35=AP for each open position. If you don't have it you will receive "no open position" in message AO instead of 35=AP message. In the positon report, you need to look at 704 (LongQty) or 705 (shortQty). If you see 704 it is long order (buy order), if you see 705 it is short order (sell order).

```
8=FIX.4.4|9=149|35=AN|34=5|49=d101968168_client1|52=20151111-21:01:12.
↪396|56=FXCM|57=U100D1|1=01958448|60=20151111-21:01:12.
↪395|263=1|581=6|710=4|715=20151111|724=0|10=085|

8=FIX.4.4|9=565|35=AP|34=8|49=FXCM|50=U100D1|52=20151111-20:19:59.929|56=d101968168_
↪client1|1=01958448|11=FIX.4.4:d101968168_client1->FXCM/U100D1-1437981786837-
↪10|15=EUR|37=207486895|55=EUR/USD|60=20150727-07:23:08|325=N|336=FXCM|526=fix_
↪example_
↪test|581=6|625=U100D1|710=4|715=20151111|721=3684204026|724=0|727=2|728=0|730=1.
↪10728|731=1|734=0|912=N|9000=1|9038=260|9040=-21.16|9041=80775478|9042=20150727-
↪07:23:08|9053=0.
↪8|453=1|448=FXCMID|447=D|452=3|802=4|523=32|803=26|523=d101968168|803=2|523=fix-
↪test112|803=22|523=1958448|803=10|702=1|703=TQ|704=10000|753=1|707=CASH|708=0|10=137|
```

## 6.5 How can I get closed positions?

First you need to send position request with 724=1

```

8=FIX.4.4|9=177|35=AN|34=6|49=d101968168_client1|52=20151111-21:01:12.
↪400|56=FXCM|57=U100D1|1=01958448|60=20151111-21:01:12.
↪400|263=1|581=6|710=5|715=20151111|724=1|9012=20150311|9014=20151112|10=110|
9043      Closing price
730      Open price
9052      PnL for the closed position
9053      Commission
9040      Interest fee associated whit this position
9038      used margin

8=FIX.4.4|9=702|35=AP|34=20|49=FXCM|50=U100D1|52=20151111-21:01:11.936|56=d101968168_
↪client1|1=01958448|11=FIX.4.4:d101968168_client1->FXCM/U100D1-1428599035518-
↪4|15=EUR|37=202027586|55=EUR/USD|60=20150519-03:30:43|325=N|336=FXCM|526=fix_
↪example_
↪test|581=6|625=U100D1|710=5|715=20151111|721=3533878441|724=1|727=13|728=0|730=1.
↪06572|731=1|734=0|912=Y|9000=1|9040=-6.08|9041=78911063|9042=20150409-
↪17:03:56|9043=1.12979|9044=20150519-03:30:43|9048=U100D1_
↪16679142D2EE08ABE053142B3C0A452A_05192015032653174913_QCV-127|9049=FXTS|9052=640.
↪7|9053=0.8|9054=204437509|453=1|448=FXCM_
↪ID|447=D|452=3|802=4|523=32|803=26|523=d101968168|803=2|523=fix-
↪test112|803=22|523=1958448|803=10|702=1|703=TQ|704=10000|753=1|707=CASH|708=0|10=042|

```

## 6.6 How can I close open positions?

If your account is **non-hedgng** account, you just need to send same quantity with opposite side.

If your account is **hedging** account, get the ticket id tag 9041 from open position then send single market order with ticket id = 9041 and opposite side.

```

20160404-06:18:07.432 : 8=FIX.4.4 9=193 35=D 34=7 49=D101546502001_client1_
↪52=20160404-06:18:07.432 56=FXCM 57=U100D1 1=01537581 11=635953582874324718_
↪38=100000 40=1 44=0.99629 54=2 55=AUD/CAD 59=1 60=20160404-06:18:07 9041=89061181_
↪10=006

20160404-06:18:07.922 : 8=FIX.4.4 9=489 35=8 34=15 49=FXCM 50=U100D1 52=20160404-
↪06:18:07.634 56=D101546502001_client1 1=01537581 6=0.99631 11=635953582874324718_
↪14=100000 15=AUD 17=819171964 31=0.99631 32=100000 37=225010828 38=100000 39=2 40=1_
↪44=0.99631 54=2 55=AUD/CAD 58=Executed 59=1 60=20160404-06:18:07 99=0 150=F 151=0_
↪211=0 336=FXCM 625=U100D1 835=0 836=0 1094=0 9000=16 9041=89061181 9050=CM 9051=F_
↪9061=0 453=1 448=FXCM ID 447=D 452=3 802=4 523=1537581 803=10 523=d101546502001_
↪803=2 523=Halpert 803=22 523=32 803=26 10=128

```

## 6.7 Sample Code in C++

### Get FXCM System Parameters: C++

```

void FixApplication::onMessage(const FIX44::TradingSessionStatus& tss, const_
↪SessionID& session_ID)
{
    int param_count = FIX::IntConvertor::convert(tss.getField(9016));

    cout << "TSS - FXCM System Parameters" << endl;

```

(continues on next page)

(continued from previous page)

```

    for(int i = 1; i <= param_count; i++){
        FIX::FieldMap map = tss.getGroupRef(1,9016);

        string param_name = map.getField(9017);
        string param_value = map.getField(9018);

        cout << param_name << " - " << param_value << endl;
    }
}

```

**Get Rollover Interest: C++**

```

void FixApplication::onMessage(const FIX44::TradingSessionStatus& tss, const_
↳SessionID& session_ID)
{
    int symbols_count = IntConvertor::convert(tss.getField(FIELD::NoRelatedSym));
    for(int i = 1; i <= symbols_count; i++) {
        FIX44::SecurityList::NoRelatedSym symbols_group;
        tss.getGroup(i,symbols_group);
        string symbol = symbols_group.getField(FIELD::Symbol);

        cout << "      Symbol -> " << symbol << endl;
        cout << "      RolloverBuy -> " << symbols_group.getField(9003) <<_
↳endl;
        cout << "      RolloverSell -> " << symbols_group.getField(9004) <<_
↳endl;
    }
}

```

**Determine Hedging Status: C++**

```

void FixApplication::onMessage(const FIX44::CollateralReport& cr, const SessionID&_
↳session_ID)
{
    FIX44::CollateralReport::NoPartyIDs group;
    cr.getGroup(1,group);
    cout << "  Parties -> " << endl;

    int number_subID = IntConvertor::convert(group.
↳getField(FIELD::NoPartySubIDs));
    for(int u = 1; u <= number_subID; u++){
        FIX44::CollateralReport::NoPartyIDs::NoPartySubIDs sub_group;
        group.getGroup(u, sub_group);

        string sub_type = sub_group.getField(FIELD::PartySubIDType);
        string sub_value = sub_group.getField(FIELD::PartySubID);
        if(sub_type == "4000"){
            // Check sub_value for position maintenance
            // Y = Hedging
            // N = No Hedging
            // 0 = Netting
        }
    }
}

```

**Subscribe to a Symbol: C++**

```

string request_ID = "EUR_USD_Request_";
FIX44::MarketDataRequest request;
request.setField(MDReqID(request_ID));
request.setField(SubscriptionRequestType(
    SubscriptionRequestType_SNAPSHOT_PLUS_UPDATES));
request.setField(MarketDepth(0));
request.setField(NoRelatedSym(1));

FIX44::MarketDataRequest::NoRelatedSym symbols_group;
symbols_group.setField(Symbol("EUR/USD"));
request.addGroup(symbols_group);

FIX44::MarketDataRequest::NoMDEntryTypes entry_types;
entry_types.setField(MDEntryType(MDEntryType_BID));
request.addGroup(entry_types);
entry_types.setField(MDEntryType(MDEntryType_OFFER));
request.addGroup(entry_types);
entry_types.setField(MDEntryType(MDEntryType_TRADING_SESSION_HIGH_PRICE));
request.addGroup(entry_types);
entry_types.setField(MDEntryType(MDEntryType_TRADING_SESSION_LOW_PRICE));
request.addGroup(entry_types);

Session::sendToTarget(request, sessionID);

```

### Subscribe to All Symbols: C++

```

void FixApplication::onMessage(const FIX44::TradingSessionStatus& tss, const_
↪SessionID& session_ID)
{
    FIX44::MarketDataRequest request;
    request.setField(MDReqID(NextRequestID()));
    request.setField(SubscriptionRequestType(
        SubscriptionRequestType_SNAPSHOT_PLUS_UPDATES));
    request.setField(MarketDepth(0));

    FIX44::MarketDataRequest::NoMDEntryTypes entry_types;
    entry_types.setField(MDEntryType(MDEntryType_BID));
    request.addGroup(entry_types);
    entry_types.setField(MDEntryType(MDEntryType_OFFER));
    request.addGroup(entry_types);
    entry_types.setField(MDEntryType(MDEntryType_TRADING_SESSION_HIGH_PRICE));
    request.addGroup(entry_types);
    entry_types.setField(MDEntryType(MDEntryType_TRADING_SESSION_LOW_PRICE));
    request.addGroup(entry_types);

    int symbols_count = IntConvertor::convert(tss.getField(FIELD::NoRelatedSym));
    request.setField(NoRelatedSym(symbols_count));
    for(int i = 1; i <= symbols_count; i++){
        FIX44::SecurityList::NoRelatedSym symbols_group_SL;
        tss.getGroup(i, symbols_group_SL);
        string symbol = symbols_group_SL.getField(FIELD::Symbol);

        FIX44::MarketDataRequest::NoRelatedSym symbols_group_MDR;
        symbols_group_MDR.setField(Symbol(symbol));
        request.addGroup(symbols_group_MDR);
    }
}

```

(continues on next page)

(continued from previous page)

```

    Session::sendToTarget(request, sessionID);
}

```

**Create Market Order: C++**

```

FIX44::NewOrderSingle order;
order.setField(FIX::ClOrdID(NextClOrdID()));
order.setField(FIX::Account(account));
order.setField(FIX::Symbol("EUR/USD"));
order.setField(FIX::Side(FIX::Side_BUY));
order.setField(FIX::TransactTime(FIX::TransactTime()));
order.setField(FIX::OrderQty(10000));
order.setField(FIX::OrdType(FIX::OrdType_MARKET));

FIX::Session::sendToTarget(order, session_id);

```

**Create Market Range Order: C++**

In the case of a market range order, we set the OrdType to StopLimit and we must set the StopPx tag. The StopPx tag indicates the worst price we are willing to get filled at; i.e., the stop.

```

FIX44::NewOrderSingle order;
order.setField(FIX::ClOrdID(NextClOrdID()));
order.setField(FIX::Account(account));
order.setField(FIX::Symbol("EUR/USD"));
order.setField(FIX::Side(FIX::Side_BUY));
order.setField(FIX::TransactTime(FIX::TransactTime()));
order.setField(FIX::OrderQty(10000));
order.setField(FIX::OrdType(FIX::OrdType_STOPLIMIT));
order.setField(FIX::StopPx(stop));

FIX::Session::sendToTarget(order, session_id);

```

**Create Entry (Pending) Order: C++**

```

FIX44::NewOrderSingle order;
order.setField(FIX::ClOrdID(NextClOrdID()));
order.setField(FIX::Account(account));
order.setField(FIX::Symbol("EUR/USD"));
order.setField(FIX::Side(FIX::Side_BUY));
order.setField(FIX::TransactTime(FIX::TransactTime()));
order.setField(FIX::OrderQty(10000));
order.setField(FIX::OrdType(FIX::OrdType_LIMIT));
order.setField(FIX::Price(price));

FIX::Session::sendToTarget(order, session_id);

```

**Create One-Cancels-Other (OCO) Order: C++**

```

FIX44::NewOrderList olist;

olist.setField(FIX::ListID(NextClOrdID()));
olist.setField(FIX::TotNoOrders(2));
olist.setField(FIX::ContingencyType(FIX::ContingencyType_ONE_CANCELS_THE_OTHER));

FIX44::NewOrderList::NoOrders stop;
stop.setField(FIX::ClOrdID(next_ClOrdID()));

```

(continues on next page)

(continued from previous page)

```

stop.setField(FIX::ListSeqNo(0));
stop.setField(FIX::ClOrdLinkID("1"));
stop.setField(FIX::Account(account));
stop.setField(FIX::Symbol(symbol));
stop.setField(FIX::Side(FIX::Side_SELL));
stop.setField(FIX::OrderQty(20000));
stop.setField(FIX::OrdType(FIX::OrdType_STOP));
stop.setField(FIX::StopPx(stop_price));
olist.addGroup(stop);

FIX44::NewOrderList::NoOrders limit;
limit.setField(FIX::ClOrdID(next_ClOrdID()));
limit.setField(FIX::ListSeqNo(1));
limit.setField(FIX::ClOrdLinkID("1"));
limit.setField(FIX::Account(account));
limit.setField(FIX::Symbol(symbol));
limit.setField(FIX::Side(FIX::Side_SELL));
limit.setField(FIX::OrderQty(20000));
limit.setField(FIX::OrdType(FIX::OrdType_LIMIT));
limit.setField(FIX::Price(limit_price));
olist.addGroup(limit);

FIX::Session::sendToTarget(olist, session_id);

```

### Create Entry with Limit and Stop (ELS) Order: C++

The entry with limit and stop is a FXCM specific contingency type that allows you to associate a stop and limit with a specific position (or market order). In this case, the ContingencyType field must be set to 101 for the ELS contingency. When the stop or limit is executed, or when you close the position, these contingent orders will be deleted automatically.

```

FIX44::NewOrderList olist;

olist.setField(FIX::ListID(next_ClOrdID()));
olist.setField(FIX::TotNoOrders(3));
olist.setField(FIX::FIELD::ContingencyType, "101");

FIX44::NewOrderList::NoOrders order;
order.setField(FIX::ClOrdID(next_ClOrdID()));
order.setField(FIX::ListSeqNo(0));
order.setField(FIX::ClOrdLinkID("1"));
order.setField(FIX::Account(account));
order.setField(FIX::Symbol(symbol));
order.setField(FIX::Side(FIX::Side_BUY));
order.setField(FIX::Symbol(symbol));
order.setField(FIX::OrderQty(10000));
order.setField(FIX::OrdType(FIX::OrdType_MARKET));
olist.addGroup(order);

FIX44::NewOrderList::NoOrders stop;
stop.setField(FIX::ClOrdID(next_ClOrdID()));
stop.setField(FIX::ListSeqNo(1));
stop.setField(FIX::ClOrdLinkID("2"));
stop.setField(FIX::Account(account));
stop.setField(FIX::Side(FIX::Side_SELL));
stop.setField(FIX::Symbol(symbol));
stop.setField(FIX::OrderQty(10000));

```

(continues on next page)



(continued from previous page)

```

stop.setField(FIX::OrdType(FIX::OrdType_STOP));
stop.setField(FIX::StopPx(stop_price));
olist.addGroup(stop);

FIX44::NewOrderList::NoOrders limit;
limit.setField(FIX::ClOrdID(next_ClOrdID()));
limit.setField(FIX::ListSeqNo(2));
limit.setField(FIX::ClOrdLinkID("2"));
limit.setField(FIX::Account(account));
limit.setField(FIX::Side(FIX::Side_SELL));
limit.setField(FIX::Symbol(symbol));
limit.setField(FIX::OrderQty(10000));
limit.setField(FIX::OrdType(FIX::OrdType_LIMIT));
limit.setField(FIX::Price(limit_price));
olist.addGroup(limit);

FIX::Session::sendToTarget(olist, session_id);

```

### Create Market Order with Trailing Stop: C++

In our example below, we use two orders with ELS contingency type (see above for details on ELS). Specifically, we send both a market order and a stop order. What makes this stop order a trailing stop is the existence of the `FXCMPegFluctuatePts(9061)` tag, which we have enumerated as `FXCM_PEG_FLUCTUATE_PTS`. This field is set to "10" which means our stop will trail the market at a rate of 10 pips.

```

FIX44::NewOrderList olist;
olist.setField(FIX::ListID(next_ClOrdID()));
olist.setField(FIX::TotNoOrders(2));
olist.setField(FIX::FIELD::ContingencyType, "101");

FIX44::NewOrderList::NoOrders order;
order.setField(FIX::ClOrdID(next_ClOrdID()));
order.setField(FIX::ListSeqNo(0));
order.setField(FIX::ClOrdLinkID("1"));
order.setField(FIX::Account(account));
order.setField(FIX::Symbol(symbol));
order.setField(FIX::Side(FIX::Side_BUY));
order.setField(FIX::OrderQty(10000));
order.setField(FIX::OrdType(FIX::OrdType_MARKET));
olist.addGroup(order);

FIX44::NewOrderList::NoOrders stop;
stop.setField(FIX::ClOrdID(next_ClOrdID()));
stop.setField(FIX::ListSeqNo(1));
stop.setField(FIX::ClOrdLinkID("2"));
stop.setField(FIX::Account(account));
stop.setField(FIX::Side(FIX::Side_SELL));
stop.setField(FIX::Symbol(symbol));
stop.setField(FIX::OrderQty(10000));
stop.setField(FIX::OrdType(FIX::OrdType_STOP));
stop.setField(FIX::StopPx(stop_price));
stop.setField(FXCM_PEG_FLUCTUATE_PTS, "10");
olist.addGroup(stop);

FIX::Session::sendToTarget(olist, session_id);

```

### Get Order Status and Executed Amount: C++

```

void FixApplication::onMessage(const FIX44::ExecutionReport& er, const SessionID&
↳session_ID)
{
    string status = er.getField(FIELD::OrdStatus);
    string execQty = er.getField(FIELD::CumQty);

    cout << "ExecutionReport ->" << endl;
    cout << "  OrderStatus: " << status << endl;
    if(status == "2" /*Filled*/ || status == "8" /*Rejected */ || status == "4" /
↳*Cancelled*/) {
        cout << "      Executed Amount: " << execQty << endl;
    }
}

```

### Request All Open Positions: C++

```

FIX44::RequestForPositions request;
request.setField(PosReqID(NextRequestID()));
request.setField(PosReqType(PosReqType_POSITIONS));

request.setField(Account(account_ID));
request.setField(SubscriptionRequestType(SubscriptionRequestType_SNAPSHOT_PLUS_
↳UPDATES));
request.setField(AccountType(
    AccountType_ACCOUNT_IS_CARRIED_ON_NON_CUSTOMER_SIDE_OF_BOOKS_AND_IS_CROSS_
↳MARGINED));
request.setField(TransactTime());
request.setField(ClearingBusinessDate());
request.setField(TradingSessionID("FXCM"));

Session::sendToTarget(request, sessionID);

```

### Request Open Positions for a Single Account: C++

```

FIX44::RequestForPositions request;
request.setField(PosReqID(NextRequestID()));
request.setField(PosReqType(PosReqType_POSITIONS));

request.setField(Account(account_ID));
request.setField(SubscriptionRequestType(SubscriptionRequestType_SNAPSHOT_PLUS_
↳UPDATES));
request.setField(AccountType(
    AccountType_ACCOUNT_IS_CARRIED_ON_NON_CUSTOMER_SIDE_OF_BOOKS_AND_IS_CROSS_
↳MARGINED));
request.setField(TransactTime());
request.setField(ClearingBusinessDate());
request.setField(TradingSessionID("FXCM"));

request.setField(NoPartyIDs(1));
FIX44::RequestForPositions::NoPartyIDs parties_group;
parties_group.setField(PartyID("FXCM ID"));
parties_group.setField(PartyIDSource('D'));
parties_group.setField(PartyRole(3));
parties_group.setField(NoPartySubIDs(1));
FIX44::RequestForPositions::NoPartyIDs::NoPartySubIDs sub_parties;
sub_parties.setField(PartySubIDType(PartySubIDType_SECURITIES_ACCOUNT_NUMBER));
sub_parties.setField(PartySubID(account_ID));

```

(continues on next page)

(continued from previous page)

```

parties_group.addGroup(sub_parties);
request.addGroup(parties_group);

Session::sendToTarget(request, sessionID);

```

**Get All Waiting Orders: C++**

```

FIX44::OrderMassStatusRequest request;
request.setField(MassStatusReqID(NextRequestID()));
request.setField(MassStatusReqType(MassStatusReqType_STATUS_FOR_ALL_ORDERS));
request.setField(Account(account_ID));
Session::sendToTarget(request, sessionID);

```

## 6.8 FXCM Custom fields

### 6.8.1 From TradingSessionStatus(h)

FXCMSymPrecision (9001)

This shows the numerical precision of the security. For example, the USD/JPY security would show a value of 3 for this field because it is quoted to 3 decimal places. AUD/USD would show a value of 5 for this field given that it is quoted to 5 decimal places.

FXCMSymPointSize (9002)

The size of the point (pip) of the security. For example, the EUR/USD security would show a value of 0.0001 for this field. This is useful for many purposes, such as calculating the profit or loss of a position in points.

FXCMSymInterestBuy (9003)

FXCMSymInterestSell (9004)

The price is in the currency of your account for the default lot size for your server. If your account is in USD and server default lot size is 10k For example for CAD/JPY 9003(FXCMSymInterestBuy) = 0.64 - you will get \$0.64 for 10k 9004(FXCMSymInterestSell) = -1.48 - you will pay \$1.48 for 10k The server default lot size you can get from same report from tags: 9017=BASE\_UNIT\_SIZE 9018=10000

You can get it also from Trading Station in Simple Dealing Rates under columns Roll S and Roll B

FXCMPProductID (9080)

FXCMPProductID distinguishes each security by its type. There are 5 types of securities: 1-Forex, 2-Index, 3-Commodity, 4-Treasury, and 5-Bullion. As an example, GBP/USD would obviously show a value of 1 for this field, but the CFD Index JPY225 would show a value of 2.

FXCMCondDistStop (9090)

The value of this field indicates the minimum distance for stop orders on an open position. The distance referred to here is the distance between your stop order price and the current market price. For example, assume you want to place a stop order on an existing buy position. Your stop order price then must meet or exceed the minimum distance from the current bid (sell) price.

FXCMCondDistLimit (9091)

The value of this field indicates the minimum distance for limit orders on an open position. The distance referred to here is the distance between your limit order price and the current market price. For example,

assume you want to place a limit order on an existing buy position. Your limit order price then must meet or exceed the minimum distance from the current Bid (Sell) price.

`FXCMCondDistEntryStop (9092)`

This field indicates the minimum distance for new stop entry (pending) orders. The distance referred to here is the distance between your stop entry order price and the current market price. For example, assume you wanted to place a stop entry order to buy. The price of this order must or exceed the minimum distance from the current Ask (Buy) price.

`FXCMCondDistEntryLimit (9093)`

This field indicates the minimum distance for new limit entry (pending) orders. The distance referred to here is the distance between your limit entry order price and the current market price. For example, assume you wanted to place a limit entry order to buy. The price of this order must or exceed the minimum distance from the current Ask (Buy) price.

`FXCMMaxQuantity (9094)`

This is the largest quantity for which you can place an order.

`FXCMMinQuantity (9095)`

This is the smallest quantity for which you can place an order. This field only applies to CFD products. The minimum trade size for Forex must be obtained from the `Quantity(53)` field from `CollateralReport`.

`FXCMTradingStatus (9096)`

This field indicates whether the trading desk is opened or closed. When trading is open, this will return “O” for Open. When trading is closed, this will return “C” for Closed. Forex securities are open throughout the entire trading week. However, CFD securities such as Indices often have daily schedules and/or daily break times. In order to determine if a CFD security is both Open and Tradeable, you must refer to the `MarketDataSnapshot` message. See `Requesting Market Data` for more on this topic.

## 6.8.2 From `MarketDataSnapshotFullRefresh(W)`

### Trading Status

Field: `QuoteType(537)`

Field: `QuoteCondition(276)`

To determine if a specific instrument is open and available for trading, you must refer to the `QuoteType(537)` and `QuoteCondition(276)` tags from this message. With `QuoteType`, a value of 0 = Indicative, and a value of 1 = Tradeable. With `QuoteCondition`, a value of “A” = Open, and a value of “B” = Closed.

## 6.8.3 From `CollateralReport(BA)`

### Minimum Order Qty. - Forex

Field: `Quantity(53)`

The value of this field represents the minimum quantity for which you can place a Forex order. This minimum quantity is specific to the trading account in the same `CollateralReport`. The `Account(1)` field can be used to obtain the `AccountID`. Most accounts are defaulted to 1,000 (Micro Lot).

## 6.9 FXCM System Parameters

The following is a list of parameter names. You will see these returned as the values for Tag 9017, `FXCMParamName`. The value of this parameter is found in Tag 9018, `FXCMParamValue`.

### `BASE_CRNCY`

This parameter shows the currency of the account. Margin, P/L, balance, and equity will all be expressed in this currency. An example base currency would be “USD”.

### `SERVER_TIME_UTC`

The value of this parameter indicates whether or not time values sent from the server will be expressed in UTC. If this is the case, the value of this parameter will be set to “UTC.” If the value of this parameter is not UTC, then time values sent from the server will be expressed in the local time zone of the server, which can be checked with the `BASE_TIME_ZONE` parameter.

### `BASE_TIME_ZONE`

This parameter shows the name of the time zone of the server. For example, “America/New\_York.”

### `COND_DIST`

This parameter shows the minimum recommended distance between the price of new stop or limit orders and the current market price. This is expressed in pips and it is generally defaulted to 0.10. It is important to note that CFD securities often have their own minimum stop or limit distances, which should be checked in the `SecurityList` message.

### `COND_DIST_ENTRY`

This parameter shows the minimum recommended distance between the price of new stop entry or limit entry orders and the current market price. This is expressed in pips and it is generally defaulted to 0.10. It is important to note that CFD securities often have their own minimum stop entry or limit entry distances, which should be checked in the `SecurityList` message.

### `BASE_UNIT_SIZE`

The minimum order size allowed for Forex securities. For example, 1,000 (Micro Lot) or 10,000 (Mini Lot). Note that in order to check the minimum order size for CFD securities, it is necessary to check the `FXCMMinQuantity` (9095) Tag from the `SecurityList` message. It is recommended that your application relies on this field when determining minimum order size for all securities, including Forex.

### `END_TRADING_DAY`

The value of this parameter contains the time when the trading day ends. The time is expressed in the format hh:mm:ss, where hh is in 24-hour format, mm is minutes, and ss is seconds. The time is always in UTC time. For example, 21:00:00.

### Disclaimer

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited (“FXCM LTD”), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the “FXCM Group”], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM

Australia Pty. Limited (“FXCM AU”) (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group’s websites prior to taking further action.”

---

## JAVA API Specifications

---

Java trading SDK, a wrapper SDK of FIX API, provides clients with a fully functioning programmable API into the FXCM FX trading platform. The API's main features are streaming executable FX trading prices, the ability to open/close positions and entry orders as well as set/update/delete stops and limits. The API Object model is based on the FIX specification for FX. It is scalable, light and robust and is compatible on any Java-compliant operating system.

### 7.1 Getting Started

- Open a demo [Trading Station II](#) account.
- Download the package [here](#).
- Documents are in the package at `trading_sdk\fxcm-api\javadoc`.
- Sample code at `trading_sdk\fxcm-api\src\QATest.java`

### 7.2 Running QATest.java

To run the program, it needs to be passed as below arguments:

```
loginid loginpwd connection_name hostUrl test_command
```

- loginid: your Trading Station username
- loginpwd: your Trading station password
- connection\_name: Demo or Real
- hostUrl: <http://www.fxcorporate.com/Hosts.jsp>

test\_command is one of the following:

LISTEN:	Just listen <b>for</b> message, do <b>not</b> do anything
CMO:	createMarketOrder (previously quoted)
SSLMO:	<b>set</b> Stop/Limit on an <b>open</b> position
USLMO:	update Stop/Limit price on a position
DSLMO:	delete Stop/Limit <b>from</b> a position
CEO:	create entry order
SSLEO:	<b>set</b> Stop/Limit on an entry order
USLEO:	update Stop/Limit on an entry order
DSLEO:	remove Stop/Limit on an entry order
DEO:	remove Entry Order
CLOSEMO:	close position
UREO:	Update rate on an entry order
MDH:	Retrieve Marke data history
RECONNECT:	Reconnect the session

## 7.3 Login

```
private void setup(IGenericMessageListener aGenericListener, boolean aPrintStatus) {
    try {
        // step 1: get an instance of IGateway from the GatewayFactory
        if (mFxcGateway == null) {
            mFxcGateway = GatewayFactory.createGateway();
        }
        /*
         step 2: register a generic message listener with the gateway, this
         listener in particular gets all messages that are related to the trading
         platform Quote,OrderSingle,ExecutionReport, etc...
        */
        mFxcGateway.registerGenericMessageListener(aGenericListener);
        mStatusListener = new DefaultStatusListener(aPrintStatus);
        mFxcGateway.registerStatusMessageListener(mStatusListener);
        if (!mFxcGateway.isConnected()) {
            System.out.println("client: login");
            FXCMLoginProperties properties = new FXCMLoginProperties(mUsername,
↪mPassword, mStation, mServer, mConfigFile);
            /*
             step 3: call login on the gateway, this method takes an instance of
↪FXCMLoginProperties
             which takes 4 parameters: username,password,terminal and server or
↪path to a Hosts.xml
             file which it uses for resolving servers. As soon as the login
↪method executes your listeners begin
             receiving asynch messages from the FXCM servers.
            */
            mFxcGateway.login(properties);
        }
        //after login you must retrieve your trading session status and get accounts
↪to receive messages
        mFxcGateway.requestTradingSessionStatus();
        mAccountMassID = mFxcGateway.requestAccounts();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



## 7.4 Rollover

With Java API you can get the current rollover for each symbol, it can be done with the functions `getFXCMSymInterestBuy()` and `getFXCMSymInterestSell()` from `TradingSecurity` Class, for Long and Short positions:

For example:

```
getFXCMSymInterestBuy() = 0.12      you will get $0.12 for 10k
getFXCMSymInterestSell() = -0.39    you will pay $0.39 for 10k
```

The 10k in this example is the server default base unit size, it can be found with `FXCMParamValue` where `FXCMParamName = "BASE_UNIT_SIZE"`

## 7.5 Sample Codes

1. RSI signal and back testing strategy
2. CCI Oscillator strategy
3. Breakout strategy
4. Range Stochastic strategy
5. Mean Reversion

**Note:** This is for personal use and abides by our [EULA](#)

For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com)

### Release Notes:

Build.number = 260: Roll up of all previous builds, plus fixes for range entry order with Good Til Date semantics;

### Disclaimer:

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action.



---

# ForexConnect API Specifications

---

The ForexConnect SDK is designed to get trading data, trade, load price histories and subscribe for the most recent prices. It is intended to be used by FXCM clients on auto-trading robots and systems, chart and market analysis application, custom trading application on FXCM accounts.

ForexConnect supports Python, C++, C#, Java, VB, VBA, Windows, Linux, iOS and Android for mobile devices. And it is free.

You can use ForexConnect on Trading station account, no extra setup required.

---

**Note:** If using O2G2 namespace, keep in mind that it is currently deprecated as it has not been updated since the beginning of 2015. It may give the users errors or not be compatible in certain cases.

---

## 8.1 Getting Started

- 1) Open a [demo Trading Station II account](#).
- 2) Download [ForexConnect SDK](#)
- 3) Examples codes and documents are included in the ForexConnectAPI packages after installation.
- 4) Online documents: [Getting Started](#)
- 5) [ForexConnect using Matlab](#)
- 6) [ForexConnect sample code for Android/iOS/macOS/Python/Linux/Windows](#)
- 7) [ForexConnect using Python](#)

## 8.2 Top Development Platform IDEs

- Windows 32bit and 64bit – [Visual Studio 2005 and up](#)

- Linux 32bit and 64bit – [Eclipse](#)
- iOS – [Xcode](#)
- Android - [Android Studio](#)

## 8.3 Table manager vs Non-table manager

Table manager preload all tables to your local memory, it is an in-memory representation of API tables. The table manager allows you to subscribe to table change events such as updates, adding rows, or removing rows. It is important to note that the SummaryTable is only accessible through the table manager. Table manager presents a performance decrease because it is constantly recalculating fields.

Non-table manager allow you to capture table updates adhoc via the use of a class that implements the IO2GResponseListener interface. It give performance advantage but you need to calculate some fields such as PipCost or P/L.

## 8.4 Request Current Balance

You need to request the table from server. Please refer to NonTableManagerSamples\PrintTable example program:

```
private static O2GAccountRow GetAccount(O2GSession session)
{
    O2GResponseReaderFactory readerFactory = session.getResponseReaderFactory();
    if (readerFactory == null)
    {
        throw new Exception("Cannot create response reader factory");
    }
    O2GLoginRules loginRules = session.getLoginRules();
    O2GResponse response = loginRules.getTableRefreshResponse(O2GTableType.Accounts);
    O2GAccountsTableResponseReader accountsResponseReader = readerFactory.
↪createAccountsTableReader(response);
    for (int i = 0; i < accountsResponseReader.Count; i++)
    {
        O2GAccountRow accountRow = accountsResponseReader.getRow(i);
        Console.WriteLine("AccountID: {0}, Balance: {1}", accountRow.AccountID, ↪
↪accountRow.Balance);
    }
    return accountsResponseReader.getRow(0);
}
```

## 8.5 Retrieve Price History

For pricehistory, you need to use non-table manager. You can see examples under NonTableManagerSamples\GetHistPrices

---

**Note:** This is for personal use and abides by our [EULA](#). For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com)

---

## 8.6 Sample Codes

### 1. Learn how to build and backtest:

- RSI Signals
- CCI Oscillator
- Breakout strategy
- Range Stochastic Strategy
- Mean Reversion Strategy

### 2. Some examples like attached stop limit to position, create if-then ELS order, get rollover

### 3. Historical data download

#### Disclaimer:

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action.



FXCM provides several sample data for free. Historical tick, candle, order flow, sentiment and volume data.

### 9.1 TickData

Enjoy free access to our historical Tick Data.

Our repository contains Tick Data from 4 January 2015. The data is compiled by trading instrument for that trading week. The files are stored in our public directory and is updated every Monday:

```
https://tickdata.fxcorporate.com/{instrument}/{year}/{int of week of year}.csv.gz
```

Instrument:

```
AUDCAD, AUDCHF, AUDJPY, AUDNZD, CADCHF, EURAUD,  
EURCHF, EURGBP, EURJPY, EURUSD, GBPCHE, GBPJPY,  
GBPNZD, GBPUSD, NZDCAD, NZDCHF, NZDJPY, NZDUSD,  
USDCAD, USDCHF, USDJPY, AUDUSD, CADJPY, GBPCAD,  
USDTRY, EURNZD
```

Year: 2015, 2016, 2017, 2018

Week: 1 to 52/53

To give an example, the path **for** extracting EURUSD data **for** the 1st week of 2015,   
↪ would be:

```
https://tickdata.fxcorporate.com/EURUSD/2015/1.csv.gz
```

If you are familiar with Python, we have two scripts that you may use for Python 2.7 and Python 3.4

## 9.1.1 Sample tick data code

### 9.1.1.1 TickData34.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 08 15:46:59 2016

@author: fxcm
"""
from StringIO import StringIO
from io import BytesIO
import gzip
import urllib
import datetime

url = 'https://tickdata.fxcorporate.com/' ##This is the base url
url_suffix = '.csv.gz' ##Extension of the file name
symbol = 'EURUSD' ##symbol we want to get tick data for
##Available Currencies
##AUDCAD,AUDCHF,AUDJPY, AUDNZD,CADCHF,EURAUD,EURCHF,EURGBP
##EURJPY,EURUSD,GBPCHF,GBPJPY,GBPNZD,GBPUSD,GBPCHF,GBPJPY
##GBPNZD,NZDCAD,NZDCHF,NZDJPY,NZDUSD,USDCAD,USDCHF,USDJPY

##The tick files are stored a compressed csv. The storage structure comes as {symbol}
↳/{year}/{week_of_year}.csv.gz
##The first week of the year will be 1.csv.gz where the
##last week might be 52 or 53. That will depend on the year.
##Once we have the week of the year we will be able to pull the correct file with the
↳data that is needed.
start_dt = datetime.date(2015,7,16)##random start date
end_dt = datetime.date(2015,8,16)##random end date

start_wk = start_dt.isocalendar()[1]##find the week of the year for the start
end_wk = end_dt.isocalendar()[1] ##find the week of the year for the end
year = str(start_dt.isocalendar()[0]) ##pull out the year of the start

###The URL is a combination of the currency, year, and week of the year.
###Example URL https://tickdata.fxcorporate.com/EURUSD/2015/29.csv.gz
###The example URL should be the first URL of this example

##This will loop through the weeks needed, create the correct URL and print out the
↳length of the file.
for i in range(start_wk, end_wk ):
    url_data = url + symbol+'/' +year+'/' +str(i)+url_suffix
    print(url_data)
    requests = urllib.request.urlopen(url_data)
    buf = BytesIO(requests.read())
    f = gzip.GzipFile(fileobj=buf)
    data = f.read()
    print(len(data))
```



### 9.1.1.2 TickData27.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 08 15:46:59 2016

@author: fxcmm
"""
from StringIO import StringIO
import gzip
import urllib2
import datetime

url = 'https://tickdata.fxcorporate.com/' ##This is the base url
url_suffix = '.csv.gz' ##Extension of the file name
symbol = 'EURUSD' ##symbol we want to get tick data for
##Available Currencies
##AUDCAD,AUDCHF,AUDJPY, AUDNZD,CADCHF,EURAUD,EURCHF,EURGBP
##EURJPY,EURUSD,GBPCHF,GBPJPY,GBPNZD,GBPUSD,GBPCHF,GBPJPY
##GBPNZD,NZDCAD,NZDCHF,NZDJPY,NZDUSD,USDCAD,USDCHF,USDJPY

##The tick files are stored a compressed csv. The storage structure comes as {symbol}
↳/{year}/{week_of_year}.csv.gz
##The first week of the year will be 1.csv.gz where the
##last week might be 52 or 53. That will depend on the year.
##Once we have the week of the year we will be able to pull the correct file with the
↳data that is needed.
start_dt = datetime.date(2015,7,16) ##random start date
end_dt = datetime.date(2015,8,16) ##random end date

start_wk = start_dt.isocalendar()[1] ##find the week of the year for the start
end_wk = end_dt.isocalendar()[1] ##find the week of the year for the end
year = str(start_dt.isocalendar()[0]) ##pull out the year of the start

###The URL is a combination of the currency, year, and week of the year.
###Example URL https://tickdata.fxcorporate.com/EURUSD/2015/29.csv.gz
###The example URL should be the first URL of this example

##This will loop through the weeks needed, create the correct URL and print out the
↳length of the file.
for i in range(start_wk, end_wk ):
    url_data = url + symbol+'/' +year+'/' +str(i)+url_suffix
    print(url_data)
    request = urllib2.Request(url_data)
    response = urllib2.urlopen(request)
    buf = StringIO(response.read())
    f = gzip.GzipFile(fileobj=buf)
    data = f.read()
    print(len(data))
```

**Note:** Losses can exceed deposits. Past performance is not indicative of future results. Timestamps are in UTC. Data points are indicative and based on the lowest spreads available exclusively on Active Trader accounts. This is for personal use and abides by our EULA. For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com).

## 9.2 CandleData

Enjoy free access to our historical Time Series or Candle Data.

Our repository contains Candle Data from 1 January 2012. The data is compiled by trading instrument for that trading week for m1 & H1, and trading year for D1. The files are stored in our public directory and is updated every Monday for minute (m1) and hour (H1) data only:

```
https://candledata.fxcorporate.com/{periodicity}/{instrument}/{year}/{int of week of ↵
↵year}.csv.gz
```

### Periodicity

m1, H1, D1

### Instrument

AUDCAD, AUDCHF, AUDJPY, AUDNZD, CADCHF, EURAUD,  
EURCHF, EURGBP, EURJPY, EURUSD, GBPCHE, GBPJPY,  
GBPNZD, GBPUSD, NZDCAD, NZDCHF, NZDJPY, NZDUSD,  
USDCAD, USDCHE, USDJPY

### Year

2012, 2013, 2014, 2015, 2016, 2017, 2018

### Week

1 to 52/53 (only applicable to m1 and H1)

To give an example, the path for extracting EURUSD minute-data for the 1st week of 2012 would be:

```
https://candledata.fxcorporate.com/m1/EURUSD/2012/1.csv.gz
```

To give an example, the path for extracting EURUSD hourly-data for the 1st week of 2012 would be:

```
https://candledata.fxcorporate.com/H1/EURUSD/2012/1.csv.gz
```

To give an example, the path for extracting EURUSD daily-data for 2012 would be:

```
https://candledata.fxcorporate.com/D1/EURUSD/2012.csv.gz
```

If you are familiar with Python, we have three scripts that you may use for Python 2.7, Python 3.4, or a pandas data frame.

### 9.2.1 Sample candle data code

#### 9.2.1.1 CandleData(pandas).py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb 08 08:11:38 2018

@author: fxcm
"""
```

(continues on next page)

(continued from previous page)

```

import datetime
import pandas as pd

url = 'https://candledata.fxcorporate.com/' ##This is the base url
periodicity='m1' ##periodicity, can be m1, H1, D1
url_suffix = '.csv.gz' ##Extension of the file name
symbol = 'EURUSD'
##symbol we want to get tick data for
##Available Currencies
##AUDCAD,AUDCHF,AUDJPY, AUDNZD,CADCHF,EURAUD,EURCHF,EURGBP
##EURJPY,EURUSD,GBPCHF,GBPJPY,GBPNZD,GBPUSD,GBPCHF,GBPJPY
##GBPNZD,NZDCAD,NZDCHF,NZDJPY,NZDUSD,USDCAD,USDCHF,USDJPY

##The candle files are stored in compressed csv. The storage structure comes as
->{periodicity}/{symbol}/{year}/{week_of_year}.csv.gz
##The first week of the year will be 1.csv.gz where the
##last week might be 52 or 53. That will depend on the year.
##Once we have the week of the year we will be able to pull the correct
##file with the data that is needed.
start_dt = datetime.date(2017,7,5) ##random start date
end_dt = datetime.date(2017,12,16) ##random end date

start_wk = start_dt.isocalendar()[1] ##find the week of the year for the start
end_wk = end_dt.isocalendar()[1] ##find the week of the year for the end
year = str(start_dt.isocalendar()[0]) ##pull out the year of the start

###The URL is a combination of the currency, periodicity, year, and week of the year.
###Example URL https://candledata.fxcorporate.com/m1/EURUSD/2017/29.csv.gz
###The example URL should be the first URL of this example
data=pd.DataFrame()
##This will loop through the weeks needed, create the correct URL and print out the
->lenght of the file.
for i in range(start_wk, end_wk ):
    url_data = url + periodicity + '/' + symbol + '/' + year + '/' + str(i) + url_
->suffix
    print(url_data)
    tempdata = pd.read_csv(url_data, compression='gzip')
    data=pd.concat([data, tempdata])
print(data)

```

### 9.2.1.2 CandleData34.py

```

# -*- coding: utf-8 -*-
"""
Created on Thu Feb 08 07:35:59 2018

@author: fxcm
"""
##from StringIO import StringIO
from io import BytesIO
import gzip
import urllib.request as ur

```

(continues on next page)

(continued from previous page)

```

import datetime

url = 'https://candledata.fxcorporate.com/' ##This is the base url
periodicity='m1' ##periodicity, can be m1, H1, D1
url_suffix = '.csv.gz' ##Extension of the file name
symbol = 'EURUSD' ##symbol we want to get candle data for
##Available Currencies
##AUDCAD,AUDCHF,AUDJPY, AUDNZD,CADCHF,EURAUD,EURCHF,EURGBP
##EURJPY,EURUSD,GBPCHF,GBPJPY,GBPNZD,GBPUSD,GBPCHF,GBPJPY
##GBPNZD,NZDCAD,NZDCHF,NZDJPY,NZDUSD,USDCAD,USDCHF,USDJPY

##The candle files are stored in compressed csv. The storage structure comes as
↳{periodicity}/{symbol}/{year}/{week_of_year}.csv.gz
##The first week of the year will be 1.csv.gz where the
##last week might be 52 or 53. That will depend on the year.
##Once we have the week of the year we will be able to pull the correct file with the
↳data that is needed.
start_dt = datetime.date(2017,7,5)##random start date
end_dt = datetime.date(2017,12,16)##random end date

start_wk = start_dt.isocalendar()[1]##find the week of the year for the start
end_wk = end_dt.isocalendar()[1] ##find the week of the year for the end
year = str(start_dt.isocalendar()[0]) ##pull out the year of the start

###The URL is a combination of the currency, periodicity, year, and week of the year.
###Example URL https://candledata.fxcorporate.com/m1/EURUSD/2017/29.csv.gz
###The example URL should be the first URL of this example

##This will loop through the weeks needed, create the correct URL and print out the
↳length of the file.
for i in range(start_wk, end_wk ):
    url_data = url + periodicity+'/' +symbol+'/' +year+'/' +str(i)+url_suffix
    print(url_data)
    requests = ur.urlopen(url_data)
    buf = BytesIO(requests.read())
    f = gzip.GzipFile(fileobj=buf)
    data = f.read()
    print(len(data))

```

### 9.2.1.3 CandleData27.py

```

# -*- coding: utf-8 -*-
"""
Created on Thu Feb 08 07:35:59 2018

@author: fxcm
"""
from StringIO import StringIO
import gzip
import urllib2
import datetime

```

(continues on next page)

(continued from previous page)

```

url = 'https://candledata.fxcorporate.com/' ##This is the base url
periodicity='m1' ##periodicity, can be m1, H1, D1
url_suffix = '.csv.gz' ##Extension of the file name
symbol = 'EURUSD' ##symbol we want to get tick data for
##Available Currencies
##AUDCAD,AUDCHF,AUDJPY, AUDNZD,CADCHF,EURAUD,EURCHF,EURGBP
##EURJPY,EURUSD,GBPCHF,GBPJPY,GBPNZD,GBPUSD,GBPCHF,GBPJPY
##GBPNZD,NZDCAD,NZDCHF,NZDJPY,NZDUSD,USDCAD,USDCHF,USDJPY

##The candle files are stored in compressed csv. The storage structure comes as
→{periodicity}/{symbol}/{year}/{week_of_year}.csv.gz
##The first week of the year will be 1.csv.gz where the
##last week might be 52 or 53. That will depend on the year.
##Once we have the week of the year we will be able to pull the correct file with the
→data that is needed.
start_dt = datetime.date(2017,7,5)##random start date
end_dt = datetime.date(2017,12,16)##random end date

start_wk = start_dt.isocalendar()[1]##find the week of the year for the start
end_wk = end_dt.isocalendar()[1] ##find the week of the year for the end
year = str(start_dt.isocalendar()[0]) ##pull out the year of the start

###The URL is a combination of the currency, periodicity, year, and week of the year.
###Example URL https://candledata.fxcorporate.com/m1/EURUSD/2017/29.csv.gz
###The example URL should be the first URL of this example

##This will loop through the weeks needed, create the correct URL and print out the
→length of the file.
for i in range(start_wk, end_wk ):
    url_data = url + periodicity + '/' + symbol + '/' + year + '/' + str(i) + url_
→suffix
    print(url_data)
    request = urllib2.Request(url_data)
    response = urllib2.urlopen(request)
    buf = StringIO(response.read())
    f = gzip.GzipFile(fileobj=buf)
    data = f.read()
    print(len(data))

```

**Note:** Losses can exceed deposits. Past performance is not indicative of future results. Timestamps are in UTC. Data points are indicative and based on the lowest spreads available exclusively on Active Trader accounts. This is for personal use and abides by our EULA. For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com).

## 9.3 Order Flow

Enjoy a free one-month sample of our historical Orders Data.

<https://sampledata.fxcorporate.com/orders/sample.csv.gz>

**Each data set would include:**

- DateTime (UTC)
- Symbol
- Quantity (Volume)
- Rate (Price)

---

**Note:** Losses can exceed deposits. Past performance is not indicative of future results. Timestamps are in UTC. Data points are indicative and based on the lowest spreads available exclusively on Active Trader accounts. This is for personal use and abides by our EULA. For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com).

---

## 9.4 Sentiment

Enjoy a free one-month sample of our historical Sentiment Data also known as SSI:

```
https://sampledata.fxcorporate.com/sentiment/{instrument}.csv.gz
```

Instrument:

```
AUDJPY, AUDUSD, CADJPY, CHFJPY, EURAUD, EURCAD, EURCHF, EURGBP,  
EURJPY, EURNOK, EURSEK, EURUSD, GBPCHF, GBPJPY, GBPUSD, NZDJPY,  
NZDUSD, USDCAD, USDCF, USDCNH, USDJPY, USDNOK, USDSEK, FRA40,  
GER30, JPN225, SPX500, UK100, US30, USOil, XAGUSD, XAUUSD
```

Each data set would include:

- DateTime (EST)
- Symbol
- Name
- Value

---

**Note:** Losses can exceed deposits. Past performance is not indicative of future results. Timestamps are in UTC. Data points are indicative and based on the lowest spreads available exclusively on Active Trader accounts. This is for personal use and abides by our EULA. For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com).

---

## 9.5 Volume

Enjoy a free one-month sample of our historical Volume Data:

```
https://sampledata.fxcorporate.com/volume/{instrument}.csv.gz
```

Instrument:

```
AUDJPY, AUDUSD, CADJPY, CHFJPY, EURAUD, EURCAD, EURCHF, EURGBP,  
EURJPY, EURNOK, EURSEK, EURUSD, GBPCHF, GBPJPY, GBPUSD, NZDJPY,  
NZDUSD, USDCAD, USDCF, USDCNH, USDJPY, USDNOK, USDSEK, FRA40,  
GER30, JPN225, SPX500, UK100, US30, USOil, XAGUSD, XAUUSD
```

Each data set would include:

- DateTime (UTC)
- Symbol
- Name
- Value

---

**Note:** Losses can exceed deposits. Past performance is not indicative of future results. Timestamps are in UTC. Data points are indicative and based on the lowest spreads available exclusively on Active Trader accounts. This is for personal use and abides by our EULA. For more information, you may contact us at [api@fxcm.com](mailto:api@fxcm.com).

---

**Disclaimer:**

CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage. 73.62% of retail investor accounts lose money when trading CFDs with this provider. You should consider whether you understand how CFDs work and whether you can afford to take the high risk of losing your money. High Risk Investment Notice: Trading Forex/CFD's on margin carries a high level of risk and may not be suitable for all investors as you could sustain losses in excess of deposits. The products are intended for retail, professional and eligible counterparty clients. For clients who maintain account(s) with Forex Capital Markets Limited ("FXCM LTD"), retail clients could sustain a total loss of deposited funds but are not subject to subsequent payment obligations beyond the deposited funds and professional clients could sustain losses in excess of deposits. Prior to trading any products offered by FXCM LTD, inclusive of all EU branches, FXCM Australia Pty. Limited, FXCM South Africa (PTY) Ltd, any affiliates of aforementioned firms, or other firms within the FXCM group of companies [collectively the "FXCM Group"], carefully consider your financial situation and experience level. If you decide to trade products offered by FXCM Australia Pty. Limited ("FXCM AU") (AFSL 309763), you must read and understand the Financial Services Guide, Product Disclosure Statement and Terms of Business. Our Forex/CFD prices are set by FXCM, are not made on an Exchange and are not governed under the Financial Advisory and Intermediary Services Act. The FXCM Group may provide general commentary which is not intended as investment advice and must not be construed as such. Seek advice from a separate financial advisor. The FXCM Group assumes no liability for errors, inaccuracies or omissions; does not warrant the accuracy, completeness of information, text, graphics, links or other items contained within these materials. Read and understand the Terms and Conditions on the FXCM Group's websites prior to taking further action."





## CHAPTER 10

---

### Support

---

For API Support, email [api@fxcm.com](mailto:api@fxcm.com)



# CHAPTER 11

---

## License

---

### MIT License

Copyright (c) 2019 Forex Capital Markets

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.