



© 2010 Forex Capital Markets LLC

FX Trading System Trading Client API v1.04

March 11, 2010

Abstract

This document provides information on the message flow of the FXCM Trading SDK

Revision changes

- V 1.00 (05/18/2005) Initial Draft
- V 1.01 (08/05/2005) Update to changes in order flow
- V 1.02 (09/06/2005) Update to include core class overview
- V 1.03 (05/11/2009) Update to incorrect negative responses
- V 1.04 (04/11/2010) Update to include TradingSessionStatus

Introduction

The FXCM Trading SDK 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 (<http://fixprotocol.org/>) and is very simple and easy to use.

1. Brief overview of core API classes – See javadocs for complete details
 - a. **IGateway**: this is the primary interface into the FXCM trading platform. It contains all the entry points into application usability.
 - b. **FXCMLoginProperties**: this class is used in the login method of IGateway and contains the properties necessary to log in.
 - c. **TradingSessionStatus**: this class provides information on the state of the market.
 - d. **CollateralReport**: represents an FXCM accounts properties at the time the message was generated. When it is a part of a batch response the RequestID can be used to match against the RequestID received from the IGateway.
 - e. **ExecutionReport**: this class represents an order's status in the system. When it is a part of a batch response the RequestID can be used to match against the RequestID received from the IGateway.
 - f. **PositionReport**: this class is used to represent a position's status in the FXCM system. When it is a part of a batch response the RequestID can be used to match against the RequestID received from the IGateway.
 - g. **ClosedPositionReport**: this class represents a closed position in the FXCM system. When it is a part of a batch response the RequestID can be used to match against the RequestID received from the IGateway.
 - h. **OrderSingle**: this class is used to send orders into the system.
 - i. **CollateralInquiryAck**: this class is the first leg of a batch response to retrieve accounts.
 - j. **RequestForPositionsAck**: this class is the first leg of a batch response to retrieve open or closed positions.
 - k. **MessageGenerator**: this class is a factory for all order types available in the API
 - l. **OrderCancelRequest**: this class is used to delete stop/limit orders.
 - m. **OrderCancelReplaceRequest**: this class is used to update entry order prices and also to update stop/limit order prices
 - n. **IGenericMessageListener**: implementations of this interface are registered with IGateway to receive application messages.
 - o. **IStatusMessageListener**: implementations of this interface are registered with IGateway to receive application status messages.

Requirements for Connection

- Applications will have a dependency on supplied fxmsg.jar, fxcm-api.jar, commons-logging.jar

Message Sequence & Explanation

Request	Response
Request Trading Session Status IGateway.requestTradingSessionStatus() *This must be the first request after logging in.	1 TradingSessionStatus
Request Accounts IGateway.requestAccounts() IGateway.requestAccounts(String aLoginID) IGateway.requestAccounts(long aFXCMacctID)	1 CollateralInquiryAck - Every request for accounts yields an ack message. 1..* CollateralReport - For every account on your login you will receive a CollateralReport
Request Open Positions IGateway.requestOpenPositions() IGateway.requestOpenPositions(String aLoginID) IGateway.requestOpenPositions(long aFXCMacctID)	1 RequestPositionsAck - Every request yields an ack message containing TotalNumPosReports in the batch. 0..* PositionReport - Cardinality dependant on ack message TotalNumPosReports
Request Closed Positions IGateway.requestClosedPositions() IGateway.requestClosedPositions(String aLoginID) IGateway.requestClosedPositions(long aFXCMacctID)	1 RequestPositionsAck - Every request yields an ack message containing TotalNumPosReports in the batch. 0..* PositionReport - Cardinality dependant on ack message TotalNumPosReports
Request Open Orders IGateway.requestOpenOrders()	1..* ExecutionReport - Cardinality dependent upon how many open orders the account has. - If there are no open orders ExecType will be REJECTED
Open Position Please see: QATest.testCreateMarketOrder() QATest.testCreateTrueMarketOrder() QATest.testOpenRangeOrder() QATest.testOrderList() QATest.testOCO() QATest.testOpenLimit() QATest.testOTO() QATest.testNetQuantity()	0..1 ExecutionReport - If the system rejects the order ExecType will be REJECTED 0..* ExecutionReport - Part of order lifecycle 0..1 PositionReport - If order successfully executed 0..1 Collateral Report - Generally you will receive an account update message directly after a position is opened
Close Position QATest.testCloseMarketOrder() QATest.testCloseTrueMarketOrder() QATest.testCloseRangeOrder() QATest.testCloseLimit()	0..1 BusinessMessageReject - If the system rejects the order else 0..* ExecutionReport - Part of order lifecycle 0..* PositionReport - Updates the state of the position if accepted 0..1 ClosedPositionReport - if order successful executed 0..1 Collateral Report - Generally you will receive an account update message directly after a position is opened
Entry Order QATest.testCreateEntryOrder()	0..1 ExecutionReport - If successful 0..1 ExecutionReport - If the system rejects the order ExecType will be REJECTED
Set Stop/Limit QATest.testSetSLEntryOrder() QATest.testSetSLMarketOrder()	0..1 ExecutionReport - If successful 0..1 ExecutionReport - If the system rejects the order ExecType will be REJECTED
Delete Stop/Limit QATest.testDeleteSLEntryOrder() QATest.testDeleteSLMarketOrder()	0..1 ExecutionReport - If successful 0..1 OrderCancelReject - If rejected
Update Stop/Limit QATest.testUpdateSLEntryOrder() QATest.testUpdateSLMarketOrder()	0..1 ExecutionReport - If successful 0..1 OrderCancelReject - If the system rejects the order ExecType will be CANCELLED
Delete Entry Order QATest.testDeleteEntryOrder()	0..1 ExecutionReport - If successful 0..1 OrderCancelReject - If the system rejects the order ExecType will be CANCELLED

Update Entry Order QATest.testUpdateRateEntryOrder()	0..1 ExecutionReport - <i>If successful</i> 0..1 OrderCancelReject - <i>If the system rejects the order ExecType will be CANCELLED</i>
---	---