
ioi-api-doc Documentation

Release 1.6.0

Terrence C. Kim

Dec 07, 2021

Contents

1	Introduction	3
1.1	Support	3
1.2	How to install serverapi.exe	4
1.3	Creating User Identities	8
1.4	Using User Identities	10
1.5	IOI API Code Samples	10
2	IOI API Subscription	11
2.1	Bloomberg API Service for IOI API Subscription	11
2.2	Accessing the Test Environment	11
2.3	API Demo Tool	12
2.4	Desktop vs. Server Authentication	12
2.5	IOI API Subscription	13
2.6	Error Message	22
2.7	Description of Elements	22
2.8	Actionable IOI	24
3	FAQ	27
3.1	General FAQ	27
4	Glossary	29
	Bibliography	31

This document is for developers who will use the Bloomberg IOI API Server to publish and subscribe to IOI (Indication of Interest) messages.

IOI API allows clients to use Bloomberg API 3.0 to automate the IOI publishing into Bloomberg and to subscribe IOI message from Bloomberg.

The Bloomberg API uses an event-driven model. The IOI API is an extension of Bloomberg API 3.0 and it lets users integrate streaming real-time and static data into their own custom applications. The user can choose the data they require down to the level of individual fields. The Bloomberg API 3.0 programming interface implementations are extremely lightweight. For details to the Desktop API, please refer to the Desktop API Programmers Guide from WAPI<GO>.

The Bloomberg API interface is thread-safe and thread-aware, giving applications the ability to utilize multiple processors efficiently. The Bloomberg API supports run-time downloadable schema for the service it provides, and it provides methods to query these schemas at runtime. This means additional service in Bloomberg API is supported without addition to the interface.

The object model for Java, .NET and C++ are identical. The C interface provides a C-style version of the object model.

Important: Due to the trading nature with the various Trading API's at Bloomberg (e.g. EMSX API, IOI API, etc.) Bloomberg cannot legally assist on the client-side coding other than providing a high-level overview of the service, advice on some of the best practices approach to use the request/response paradigm and asynchronous event-driven nature of the subscription paradigm.

It is highly recommended that the technical resource working on the Trading API has extensive programming experiences and a solid understanding of software application architecture.

Warning: Please note that performance/load test should never be performed on any of the API environment as this is a shared environment and we monitor and increase capacity as needed.

CHAPTER 1

Introduction

The IOI API Server provides Bloomberg users with the ability to publish IOIs into Bloomberg and subscribe IOIs in Equities and Derivatives using Bloomberg API

The IOI API Server for subscribing requires separate authorization by the publishing broker on top of the Bloomberg authorization.

Note: IOI API users will need the following steps completed before using the IOI API service.

1. Signed EMSxNET Order Originator Agreement.
 2. Install serverapi.exe and register with Bloomberg.
 3. Enable IOI API per UUID by the Global EMSX Trade Desk for Test (Beta) and Production.
 4. To get access to IOI API in UAT and production, please click <Help><Help> on EMSX<GO>.
 5. Download Bloomberg Desktop API v3 SDK from WAPI<GO> in Bloomberg terminal.
-

To get access to IOI API in UAT and production, please click <Help><Help> on EMSX<GO>.

1.1 Support

For all IOI functionality and IOI API technical inquiries please contact the EMSX Trade Desk. They are available 24/6 and please ensure you provide your Bloomberg UUID.

By Bloomberg Terminal:

HELP 2x (F1 key) on your terminal, ask to speak to the EMSX Trade Desk HELP 1x (F1 key) on your terminal, to compose an email message to the EMSX Trade Desk.

By email:

emsx@bloomberg.net

By Phone:

Please call your local global customer support number and ask to speak to the EMSX Trade Desk

+1-212-617-2000	+44-20-7330-7500	+65-6212-1000
-----------------	------------------	---------------

1.2 How to install serverapi.exe

Please follow the following steps to install and register the installer with Bloomberg Enterprise Solutions with the assistance from EMSX Implementation team.

The following example is based on the windows environment and thus there will be slight variations when installing from linux or solaris operating system.

- Run serverapi.exe

```
C:\temp>serverapiinstaller.exe
```

- You will see the following message

```
C:\temp>serverapiinstaller.exe
logging to C:\temp\install.2016102610.152444.log

Bloomberg ECD Installer for Windows (32-bit)
Version 3.2.2.0

Warning: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of
it, may result in severe civil and criminal penalties, and will be prosecuted
to the maximum extent possible under law.

logging to C:\temp\install.2016102610.152444.log

Bloomberg ECD Installer for Windows (32-bit)
Version 3.2.2.0

Warning: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of
it, may result in severe civil and criminal penalties, and will be prosecuted
to the maximum extent possible under law.

Would you like to continue? (Y/N):
```

- Type:- Y

```
Would you like to continue? (Y/N): y

Checking connectivity to Bloomberg ...

a) via Bloomberg Network to [ Hostname = 208.134.161.62 Port = 8194 ] ...
```

(continues on next page)

(continued from previous page)

```

Succeeded.

[ Hostname = 208.134.161.158 Port = 8194 ] ...
Succeeded.

[ Hostname = 208.134.161.18 Port = 8194 ] ...
Succeeded.

[ Hostname = 208.134.161.179 Port = 8194 ] ...
Succeeded.

b) via the Internet to [ Hostname = api1.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api2.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api3.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api4.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api5.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api6.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api7.bloomberg.net Port = 8194 ] ...
Succeeded.

[ Hostname = api8.bloomberg.net Port = 8194 ] ...
Succeeded.

Which of the above routes will you use to connect to Bloomberg? (a/b):

```

- Select the appropriate network option (private vs. public/internet)

```

Which of the above routes will you use to connect to Bloomberg? (a/b): b

Bloomberg Network connectivity unavailable. Connecting via the Internet.

Select Product Class
1) blpddm    Software that provides development access to distribute data locally or
↳ contribute data to Bloomberg.
2) ServerApi Provides access to Bloomberg real-time streaming and static data
0) Quit

Please enter selection:

```

- Select:- 2 for Server API

```

Please enter selection: 2

```

(continues on next page)

(continued from previous page)

```
Installation path:
    'C:\'

Use this path? (Y/N/Q):
```

- Select:- Y

```
Use this path? (Y/N/Q): y

Downloading latest installer ...
done.

logging to C:\temp\install.2016102610.152444.log

Beginning new install ...
```

Note: If the default port is already being used by a different service it may show the following message:

```
*** WARNING: Port conflict detected with other service.
The port of the Desktop will conflict with the ServerApi should installation proceed.
↪If you still want to install
ServerApi, you will need to specify a different port number.

Do you want to continue with the installation? (Y/N) [N]:
```

- Select:- Y and enter the port

```
Do you want to continue with the installation? (Y/N) [N]:y
Please enter ServerApi listen port: [8294]:8294
```

- Select the version:-

```
Versions available for ServerApi
1) 3.46.6.0      Windows      ServerAPI 2014-07
2) 3.48.8.1      Windows      ServerAPI 2014-09
3) 3.48.9.1      Windows      ServerAPI 2014-11
4) 3.50.7.1      Windows      ServerAPI 2015-01
5) 3.56.4.1      Windows      ServerAPI 2015-04
6) 3.60.0.1      Windows      ServerAPI 2015-07
7) 3.64.5.1      Windows      ServerAPI 2015-10
8) 3.70.0.1      Windows      ServerAPI 2016-01
9) 3.72.2.1      Windows      ServerAPI 2016-04
10) 3.82.3.1     Windows      ServerAPI 2016-10
11) 3.46.6.0     Windows64    ServerAPI 2014-07
```

(continues on next page)

(continued from previous page)

```

12) 3.48.8.1      Windows64  ServerAPI 2014-09
13) 3.48.9.1      Windows64  ServerAPI 2014-11
14) 3.50.7.1      Windows64  ServerAPI 2015-01
15) 3.56.4.1      Windows64  ServerAPI 2015-04
16) 3.60.0.1      Windows64  ServerAPI 2015-07
17) 3.64.5.1      Windows64  ServerAPI 2015-10
18) 3.70.0.1      Windows64  ServerAPI 2016-01
19) 3.72.2.1      Windows64  ServerAPI 2016-04
20) 3.82.3.1      Windows64  ServerAPI 2016-10

0) Quit
Please enter version of ServerApi that you want to install:

```

- Select the latest:-

```

Please enter version of ServerApi that you want to install: 20
Downloading ServerApi components ...

```

- Enter other information:-

Enter the following information:

Country (e.g., USA): State (e.g., NY): City or Town (e.g., New York): Company Name (e.g., Bloomberg L.P.): Department Name (e.g., Equity Trading):

- Finished:-

```

Enter the following information:

Country (e.g., USA): USA
State (e.g., NY): NY
City or Town (e.g., New York): New York
Company Name (e.g., Bloomberg L.P.): Bloomberg LP
Department Name (e.g., Equity Trading): EMSX

Creating certificate ...
done.

Registering server ...
done.

Do you want to install ServerApi as a Windows Service? (Y/N): y

Installing ServerApi as a windows Service...
service ServerApi configured for restart on first error
done
done.

```

(continues on next page)

(continued from previous page)

```
*** Please reboot your computer for changes to take effect ***
```

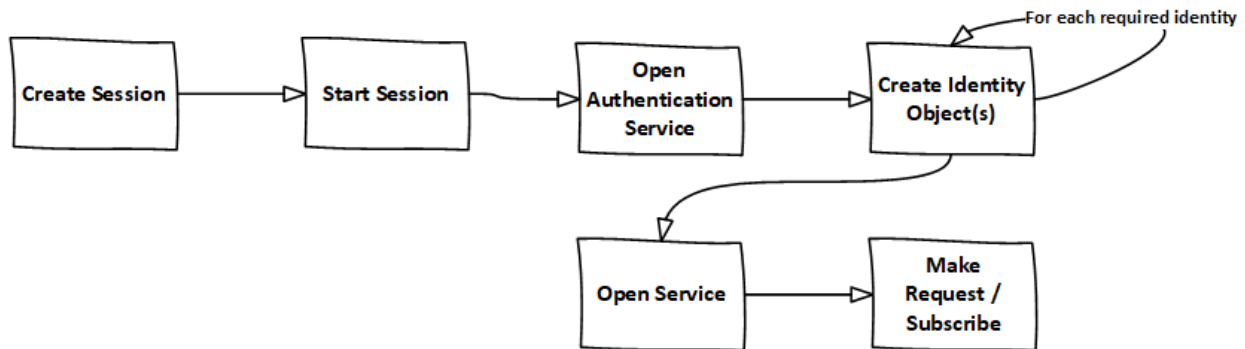
Call Bloomberg's Global Customer Support at +1 (212) 318-2000 and ask for the Global Installs desk. The Bloomberg representative will ask you to read your registration number over the phone four characters at a time.

Your registration key is:
321c-5ad5-7fa8-2954-1930-abb0-b64c-ecaf-1505-64d4

Note: Once the registration process is completed. EMSX Implementation team globally will assist with configuring the Server Side EMSX API with various execution destinations per client request.

1.3 Creating User Identities

In the server environment, the user identities must be created and cached prior to the making requests. Therefore, the process would look as follows:-



Note: Note: The IOI API Server code samples demonstrate how to create identity object.

The first new step is to open the authentication service. This is done in the same way as for any other service in the Bloomberg API. For example:-

```
d_authsvc = "//blp/apiauth";
session.openServiceAsync(d_authsvc);
```

Once the service is opened, we need to create and send an authorization request. To create an identity for a specific user, you will need the AuthID for the user. This is the name the user is known by in the EMRS system for your server. The values for these names will have been agreed with you as part of the implementation of the server, or subsequently when adding a new user. Also, an IP address is required. The only requirement for this IP address is that it is unique amongst all the identities generated for a session. You can create and send the request as follows:-

```
private Identity userIdentity;
```

(continues on next page)

(continued from previous page)

```

*
*
*
Service authService = session.getService(d_authsvc);
Request authReq = authService.createAuthorizationRequest();

authReq.set("authId", authID);
authReq.set("ipAddress", appIP);

userIdentity = session.createIdentity();

authRequestID = new CorrelationID();

try
{
    session.sendAuthorizationRequest(authReq, userIdentity, authRequestID);
}
catch (Exception e)
{
    System.out.println("Unable to send authorization request: " + e.getMessage());
}

```

In the above code, you can see that an empty identity object is created using `session.createIdentity()`. This is the object that will be populated once successful authentication has been achieved, and it is the object that will need to be cached.

We will receive a Response event for the Authentication service. In the example below, we use a `CorrelationID` to identify messages from the Authentication service, and check for success or failure:-

```

    if(msg.correlationID()==authRequestID) {

        if(msg.messageType().equals(AUTHORIZATION_SUCCESS)) {
            System.out.println("Authorised...Opening IOI service...");
            System.out.println("Seat Type: " + userIdentity.seatType().
→toString());
            session.openServiceAsync(d_service);
        } else if(msg.messageType().equals(AUTHORIZATION_FAILURE)) {
            System.out.println("Authorisation failed...");
            System.out.println(msg.toString());
            wait(1000);
            // Automatically retry...
            sendAuthRequest(session);
        } else {
            System.out.println("Unexpected authorisation message...");
            System.out.println(msg.toString());
        }
    }
}

```

When we receive the successful authorization, we can continue with opening the IOI service. If multiple authorization requests have been sent, for a number of different UUIDs, it is necessary to wait for all the responses before being able to use all the identity objects.

In the above code, you will see that we examine the ‘seatType’ of the identity. The seat type in this case will be either BPS or non-BPS.

1.4 Using User Identities

In the server environment, there is no Bloomberg terminal, and therefore no implied user can be identified. Moreover, the server is capable of connecting to any number of IOI user identities, simultaneously. Therefore, the application making the call must indicate which user is the intended target. This is done through the creation and use of Identity object.

An Identity object represents a specific Bloomberg UUID. Once created, an Identity object can be cached for 24hrs, and used with every `sendRequest()` and `subscribe()` call.

Identity objects are live, that is they remain connected to Bloomberg in real-time and are capable of receiving events. We recommend that an identity is recreated every 24hrs, to ensure that it picks up the latest changes to any user settings, including access to IOI<GO>.

Migrating the existing desktop application call to a server application simply involves changing all `sendRequest()` and `subscribe()` calls to include the appropriate identity, as follows:-

```
DAPI:
    session.sendRequest(request, requestID);
    session.subscribe(subscriptions);

Server:
    session.sendRequest(request, Identity, requestID);
    session.subscribe(subscriptions, Identity);
```

1.5 IOI API Code Samples

Important: The latest IOI API Code samples can be found [here](#).

CHAPTER 2

IOI API Subscription

This document is for developers who will use the Bloomberg IOI API Server to subscribe to IOI (Indication of Interest) messages from various brokers.

The subscribed message can be found in `IOI<GO>` function in the Bloomberg terminal.

The service name is `//blp/ioisub` for production and `//blp/ioisub-beta` to call the test environment.

Unlike the EMSX API, the IOI API service supports partial response messages which will return messages that are a subset of the information.

IOI API Subscription Schema

2.1 Bloomberg API Service for IOI API Subscription

For IOI subscription, this is accomplished by referencing `//blp/ioisub-beta` as the service name in your program. This command will allow your service to redirect all IOI API subscription to the test environment.

Once the client has thoroughly tested the custom-built strategies, they can access the production environment by changing the service name from `//blp/ioisub-beta` to `//blp/ioisub`.

2.2 Accessing the Test Environment

Bloomberg provides a test environment for clients to build and test their strategies using the IOI API.

Inside the Bloomberg Terminal type `UAT ON <GO>`. This command allows the particular terminal window and launchpad to log into the beta environment. Please note, when a user is remote into the beta environment it only affects that particular terminal window and the other Bloomberg panels will not be affected by the `UAT ON <GO>` command.

To check which environment your current view is in, type `VSAT <GO>` inside the Bloomberg terminal.

To get back to production type `UAT OFF <GO>`. Please note that the testing environment in Beta will not operate in the exact same way as the production environment. Also, please note that the beta environment is a lot slower than the production environment and no one should perform any volume or load testing in the beta environment.

2.3 API Demo Tool

API Demo Tool is a handy tool while developing on any Bloomberg API services. The API Demo Tool provides real-time schema viewing tool among other handy tools that can be leveraged during the initial development.

The API Demo Tool can be downloaded from the Bloomberg terminal along with other generic Bloomberg API code samples.

```
WAPI<GO> >> API Download Center >> Download
```

2.4 Desktop vs. Server Authentication

Desktop:

```
d_ioi = "//blp/ioiapi-beta-request"
d_host = "localhost"
d_port = 1234
```

Server:

```
d_ioi = "//blp/ioiapi-beta-request"
d_auth = "//blp/apiauth"
d_host = "abc.com"
d_port = 1234
d_user = "myAuthID"
d_ip = "10.20.30.40"
```

Set authorization request:

```
def sendAuthRequest(self, session):

    authService = session.getService(d_auth)
    authReq = authService.createAuthorizationRequest()
    authReq.set("emrsID", d_user)
    authReq.set("ipAdress", d_ip)
    self.identity = session.createIdentity()

    print ("Sending authorization rquest: %s" % (authReq))

    session.sendAuthorizationRequest(authReq, self.identity)

    print ("Authorization request sent.")

...

def processAuthorizationStatusEvent(self, event):

    print ("Processing AUTHORIZATION_STATUS event")

    for msg in event:
```

(continues on next page)

(continued from previous page)

```

        print("AUTHORIZATION_STATUS message: %s" % (msg))
...
def processEvent(self, event, session):
    try:
        ...

    elif event.eventType() == blpapi.Event.AUTHORIZATION_STATUS:
        self.processAuthorizationStatusEvent(event)
    ...

```

2.5 IOI API Subscription

The IOI API Subscription allows IOI messages over subscription service.

Full code sample:-

Subscribe IOI cs	Subscribe IOI py
Subscribe IOI java	

Hint: Please right click on the top code sample link to open in a new tab.

```

1  def createIOISubscription(self, session):
2
3      print("Create IOI subscription")
4
5      #Create the topic string for the ioi subscription.
6      ioiTopic = d_ioi + "/ioi"
7
8      subscriptions = blpapi.SubscriptionList()
9
10     subscriptions.add(topic=ioiTopic, correlationId=ioiSubscriptionID)
11
12     print("Sending subscription...")
13     session.subscribe(subscriptions)

```

Output:-

```

C:\Users\_scripts>py -3 py_dapi_SubscribeIOI.py
Bloomberg - IOI API Example - DesktopAPI - SubscribeIOI
Connecting to localhost:8194
Processing SESSION_STATUS event
SessionConnectionUp = {
    server = "localhost:8194"
    encryptionStatus = "Clear"
}

```

(continues on next page)

(continued from previous page)

```

Session connection is up
Processing SESSION_STATUS event
SessionStarted = {
    initialEndpoints[] = {
        initialEndpoints = {
            address = "localhost:8194"
        }
    }
}

Session started...
Processing SERVICE_STATUS event
ServiceOpened = {
    serviceName = "//blp/ioisub-beta"
}

IOI API service opened... Sending request...
Create IOI subscription
Sending subscription: //blp/ioisub-beta/ioi
Processing SUBSCRIPTION_STATUS event
SUBSCRIPTION_STATUS MESSAGE: SubscriptionStarted = {
    exceptions[] = {
    }
    streamIds[] = {
        "1"
    }
    receivedFrom = {
        address = "localhost:8194"
    }
    reason = "Subscriber made a subscription"
}

IOI API subscription started...
SUBSCRIPTION_STATUS MESSAGE: SubscriptionStreamsActivated = {
    streams[] = {
        streams = {
            id = "1"
            endpoint = {
                address = "localhost:8194"
            }
        }
    }
    reason = "Subscriber made a subscription"
}

Processing SUBSCRIPTION_DATA event
Ioidata = {
    ioi_instrument_type = "stock"
    ioi_instrument_stock_security_ticker = "VOD LN Equity"
    ioi_goodUntil = 2020-01-23T14:45:59.000+00:00
    ioi_bid_price_type = "fixed"
    ioi_bid_price_fixed_price = 226.500000
    ioi_bid_price_fixed_currency = "GBP"
    ioi_bid_size_type = "quantity"
    ioi_bid_size_quantity = 1000
    ioi_bid_notes = "bid notes"
    ioi_bid_qualifiers_count = 0

```

(continues on next page)

(continued from previous page)

```

    ioi_bid_natural = 0
    ioi_sentTime = 2020-01-23T14:30:59.267+00:00
    ioi_id = "-1034576088"
    state = "New"
    id_value = "-1034576088"
    originalId_value = "-1034576088"
    trader_uuid = 6767714
    trader_acronym = "B:BLP"
    trader_username = "TKIM"
    receivedTime = 2020-01-23T14:30:59.267+00:00
}

```

```

IOI MESSAGE: CorrelationID(1)
ioi_instrument_type: stock
ioi_instrument_option_legs_count: 0
ioi_instrument_option_legs_0_strike: 0
ioi_instrument_option_legs_0_expiry:
ioi_instrument_option_legs_0_type:
ioi_instrument_option_legs_0_ratio: 0
ioi_instrument_option_legs_0_underlying_type:
ioi_instrument_option_legs_0_underlying_ticker:
ioi_instrument_option_legs_0_underlying_figi:
ioi_instrument_option_legs_0_exchange:
ioi_instrument_option_legs_0_style:
ioi_instrument_option_legs_0_futureRefDate:
ioi_instrument_option_legs_0_delta: 0
ioi_instrument_option_legs_1_strike: 0
ioi_instrument_option_legs_1_expiry:
ioi_instrument_option_legs_1_type:
ioi_instrument_option_legs_1_ratio: 0
ioi_instrument_option_legs_1_underlying_type:
ioi_instrument_option_legs_1_underlying_ticker:
ioi_instrument_option_legs_1_underlying_figi:
ioi_instrument_option_legs_1_exchange:
ioi_instrument_option_legs_1_style:
ioi_instrument_option_legs_1_futureRefDate:
ioi_instrument_option_legs_1_delta: 0
ioi_instrument_option_legs_2_strike: 0
ioi_instrument_option_legs_2_expiry:
ioi_instrument_option_legs_2_type:
ioi_instrument_option_legs_2_ratio: 0
ioi_instrument_option_legs_2_underlying_type:
ioi_instrument_option_legs_2_underlying_ticker:
ioi_instrument_option_legs_2_underlying_figi:
ioi_instrument_option_legs_2_exchange:
ioi_instrument_option_legs_2_style:
ioi_instrument_option_legs_2_futureRefDate:
ioi_instrument_option_legs_2_delta: 0
ioi_instrument_option_legs_3_strike: 0
ioi_instrument_option_legs_3_expiry:
ioi_instrument_option_legs_3_type:
ioi_instrument_option_legs_3_ratio: 0
ioi_instrument_option_legs_3_underlying_type:
ioi_instrument_option_legs_3_underlying_ticker:
ioi_instrument_option_legs_3_underlying_figi:
ioi_instrument_option_legs_3_exchange:
ioi_instrument_option_legs_3_style:

```

(continues on next page)

(continued from previous page)

```
ioi_instrument_option_legs_3_futureRefDate:
ioi_instrument_option_legs_3_delta: 0
ioi_instrument_option_structure:
ioi_instrument_stock_security_ticker: VOD LN Equity
ioi_instrument_stock_security_figi:
ioi_goodUntil: 2020-01-23T14:45:59.000+00:00
ioi_bid_price_type: fixed
ioi_bid_price_fixed_price: 226
ioi_bid_price_fixed_currency: GBp
ioi_bid_price_pegged_offsetAmount: 0
ioi_bid_price_pegged_offsetFrom:
ioi_bid_price_pegged_limitPrice: 0
ioi_bid_price_reference:
ioi_bid_price_moneyness: 0
ioi_bid_size_type: quantity
ioi_bid_size_quantity: 1000
ioi_bid_size_quality:
ioi_bid_referencePrice_price: 0
ioi_bid_referencePrice_currency:
ioi_bid_volatility: 0
ioi_bid_notes: bid notes
ioi_bid_qualifiers_count: 0
ioi_bid_qualifiers_0:
ioi_bid_qualifiers_1:
ioi_bid_qualifiers_2:
ioi_bid_qualifiers_3:
ioi_bid_qualifiers_4:
ioi_offer_price_type:
ioi_offer_price_fixed_price: 0
ioi_offer_price_fixed_currency:
ioi_offer_price_pegged_offsetAmount: 0
ioi_offer_price_pegged_offsetFrom:
ioi_offer_price_pegged_limitPrice: 0
ioi_offer_price_reference:
ioi_offer_price_moneyness: 0
ioi_offer_size_type:
ioi_offer_size_quantity: 0
ioi_offer_size_quality:
ioi_offer_referencePrice_price: 0
ioi_offer_referencePrice_currency:
ioi_offer_volatility: 0
ioi_offer_notes:
ioi_offer_qualifiers_count: 0
ioi_offer_qualifiers_0:
ioi_offer_qualifiers_1:
ioi_offer_qualifiers_2:
ioi_offer_qualifiers_3:
ioi_offer_qualifiers_4:
ioi_routing_strategy_name:
ioi_routing_strategy_brief:
ioi_routing_strategy_detailed:
ioi_routing_customId:
ioi_routing_broker:
ioi_sentTime: 2020-01-23T14:30:59.267+00:00
change:
Processing SUBSCRIPTION_DATA event
Processing SUBSCRIPTION_DATA event
```

(continues on next page)

(continued from previous page)

```
Exception: raw write() returned invalid length 72 (should have been between 0 and 36)
Processing SUBSCRIPTION_DATA event
```

```
Ioidata = {
    ioi_instrument_type = "stock"
    ioi_instrument_stock_security_ticker = "SINA UW Equity"
    ioi_goodUntil = 2020-01-23T14:36:15.000+00:00
    ioi_bid_price_type = "fixed"
    ioi_bid_price_fixed_price = 40.390000
    ioi_bid_price_fixed_currency = "USD"
    ioi_bid_size_type = "quantity"
    ioi_bid_size_quantity = 8279
    ioi_bid_notes = "1/22/2020 3:38:35 PM"
    ioi_bid_qualifiers_count = 0
    ioi_bid_natural = 0
    ioi_sentTime = 2020-01-23T14:31:15.183+00:00
    ioi_id = "D4291-23JAN2020-229-x0"
    state = "New"
    id_value = "1781985133"
    originalId_value = "1781985133"
    trader_uuid = 12624540
    trader_acronym = "B:BLP"
    receivedTime = 2020-01-23T14:31:15.183+00:00
}
```

```
IOI MESSAGE: CorrelationID(1)
ioi_instrument_type: stock
ioi_instrument_option_legs_count: 0
ioi_instrument_option_legs_0_strike: 0
ioi_instrument_option_legs_0_expiry:
ioi_instrument_option_legs_0_type:
ioi_instrument_option_legs_0_ratio: 0
ioi_instrument_option_legs_0_underlying_type:
ioi_instrument_option_legs_0_underlying_ticker:
ioi_instrument_option_legs_0_underlying_figi:
ioi_instrument_option_legs_0_exchange:
ioi_instrument_option_legs_0_style:
ioi_instrument_option_legs_0_futureRefDate:
ioi_instrument_option_legs_0_delta: 0
ioi_instrument_option_legs_1_strike: 0
ioi_instrument_option_legs_1_expiry:
ioi_instrument_option_legs_1_type:
ioi_instrument_option_legs_1_ratio: 0
ioi_instrument_option_legs_1_underlying_type:
ioi_instrument_option_legs_1_underlying_ticker:
ioi_instrument_option_legs_1_underlying_figi:
ioi_instrument_option_legs_1_exchange:
ioi_instrument_option_legs_1_style:
ioi_instrument_option_legs_1_futureRefDate:
ioi_instrument_option_legs_1_delta: 0
ioi_instrument_option_legs_2_strike: 0
ioi_instrument_option_legs_2_expiry:
ioi_instrument_option_legs_2_type:
ioi_instrument_option_legs_2_ratio: 0
ioi_instrument_option_legs_2_underlying_type:
ioi_instrument_option_legs_2_underlying_ticker:
ioi_instrument_option_legs_2_underlying_figi:
ioi_instrument_option_legs_2_exchange:
```

(continues on next page)

(continued from previous page)

```

ioi_instrument_option_legs_2_style:
ioi_instrument_option_legs_2_futureRefDate:
ioi_instrument_option_legs_2_delta: 0
ioi_instrument_option_legs_3_strike: 0
ioi_instrument_option_legs_3_expiry:
ioi_instrument_option_legs_3_type:
ioi_instrument_option_legs_3_ratio: 0
ioi_instrument_option_legs_3_underlying_type:
ioi_instrument_option_legs_3_underlying_ticker:
ioi_instrument_option_legs_3_underlying_figi:
ioi_instrument_option_legs_3_exchange:
ioi_instrument_option_legs_3_style:
ioi_instrument_option_legs_3_futureRefDate:
ioi_instrument_option_legs_3_delta: 0
ioi_instrument_option_structure:
ioi_instrument_stock_security_ticker: SINA UW Equity
ioi_instrument_stock_security_figi:
ioi_goodUntil: 2020-01-23T14:36:15.000+00:00
ioi_bid_price_type: fixed
ioi_bid_price_fixed_price: 40
ioi_bid_price_fixed_currency: USD
ioi_bid_price_pegged_offsetAmount: 0
ioi_bid_price_pegged_offsetFrom:
ioi_bid_price_pegged_limitPrice: 0
ioi_bid_price_reference:
ioi_bid_price_moneyness: 0
ioi_bid_size_type: quantity
ioi_bid_size_quantity: 8279
ioi_bid_size_quality:
ioi_bid_referencePrice_price: 0
ioi_bid_referencePrice_currency:
ioi_bid_volatility: 0
ioi_bid_notes: 1/22/2020 3:38:35 PM
ioi_bid_qualifiers_count: 0
ioi_bid_qualifiers_0:
ioi_bid_qualifiers_1:
ioi_bid_qualifiers_2:
ioi_bid_qualifiers_3:
ioi_bid_qualifiers_4:
ioi_offer_price_type:
ioi_offer_price_fixed_price: 0
ioi_offer_price_fixed_currency:
ioi_offer_price_pegged_offsetAmount: 0
ioi_offer_price_pegged_offsetFrom:
ioi_offer_price_pegged_limitPrice: 0
ioi_offer_price_reference:
ioi_offer_price_moneyness: 0
ioi_offer_size_type:
ioi_offer_size_quantity: 0
ioi_offer_size_quality:
ioi_offer_referencePrice_price: 0
ioi_offer_referencePrice_currency:
ioi_offer_volatility: 0
ioi_offer_notes:
ioi_offer_qualifiers_count: 0
ioi_offer_qualifiers_0:
ioi_offer_qualifiers_1:

```

(continues on next page)

(continued from previous page)

```

ioi_offer_qualifiers_2:
ioi_offer_qualifiers_3:
ioi_offer_qualifiers_4:
ioi_routing_strategy_name:
ioi_routing_strategy_brief:
ioi_routing_strategy_detailed:
ioi_routing_customId:
ioi_routing_broker:
ioi_sentTime: 2020-01-23T14:31:15.183+00:00
change:
Processing SUBSCRIPTION_DATA event
Ioidata = {
    ioi_instrument_type = "option"
    ioi_instrument_option_legs_count = 2
    ioi_instrument_option_legs_0_strike = 230.000000
    ioi_instrument_option_legs_0_expiry = 2020-01-31T12:00:00.000+00:00
    ioi_instrument_option_legs_0_type = "Call"
    ioi_instrument_option_legs_0_ratio = 1.000000
    ioi_instrument_option_legs_0_underlying_figi = "BBG000C6K6G9"
    ioi_instrument_option_legs_0_exchange = "LN"
    ioi_instrument_option_legs_0_style = "European"
    ioi_instrument_option_legs_1_strike = 240.000000
    ioi_instrument_option_legs_1_expiry = 2020-01-31T12:00:00.000+00:00
    ioi_instrument_option_legs_1_type = "Call"
    ioi_instrument_option_legs_1_ratio = -1.250000
    ioi_instrument_option_legs_1_underlying_figi = "BBG000C6K6G9"
    ioi_instrument_option_legs_1_exchange = "LN"
    ioi_instrument_option_legs_1_style = "European"
    ioi_instrument_option_structure = "CallSpread"
    ioi_goodUntil = 2020-01-23T14:46:21.192+00:00
    ioi_bid_price_type = "fixed"
    ioi_bid_price_fixed_price = 83.630000
    ioi_bid_price_fixed_currency = ""
    ioi_bid_size_type = "quantity"
    ioi_bid_size_quantity = 1000
    ioi_bid_referencePrice_price = 202.150000
    ioi_bid_referencePrice_currency = "GBp"
    ioi_bid_notes = "bid notes"
    ioi_bid_qualifiers_count = 0
    ioi_offer_price_type = "fixed"
    ioi_offer_price_fixed_price = 83.640000
    ioi_offer_price_fixed_currency = ""
    ioi_offer_size_type = "quantity"
    ioi_offer_size_quantity = 2000
    ioi_offer_referencePrice_price = 202.150000
    ioi_offer_referencePrice_currency = "GBp"
    ioi_offer_notes = "offer notes"
    ioi_offer_qualifiers_count = 0
    state = "New"
    id_value = "0678b50e-287c-416f-813a-bf34cde3f300"
    originalId_value = "0678b50e-287c-416f-813a-bf34cde3f300"
    trader_uuid = 6767714
    trader_acronym = "S:BLP"
    receivedTime = 2020-01-23T14:31:21.577+00:00
}
IOI MESSAGE: CorrelationID(1)

```

(continues on next page)

(continued from previous page)

```
ioi_instrument_type: option
ioi_instrument_option_legs_count: 2
ioi_instrument_option_legs_0_strike: 230
ioi_instrument_option_legs_0_expiry: 2020-01-31T12:00:00.000+00:00
ioi_instrument_option_legs_0_type: Call
ioi_instrument_option_legs_0_ratio: 1
ioi_instrument_option_legs_0_underlying_type:
ioi_instrument_option_legs_0_underlying_ticker:
ioi_instrument_option_legs_0_underlying_figi: BBG000C6K6G9
ioi_instrument_option_legs_0_exchange: LN
ioi_instrument_option_legs_0_style: European
ioi_instrument_option_legs_0_futureRefDate:
ioi_instrument_option_legs_0_delta: 0
ioi_instrument_option_legs_1_strike: 240
ioi_instrument_option_legs_1_expiry: 2020-01-31T12:00:00.000+00:00
ioi_instrument_option_legs_1_type: Call
ioi_instrument_option_legs_1_ratio: -1
ioi_instrument_option_legs_1_underlying_type:
ioi_instrument_option_legs_1_underlying_ticker:
ioi_instrument_option_legs_1_underlying_figi: BBG000C6K6G9
ioi_instrument_option_legs_1_exchange: LN
ioi_instrument_option_legs_1_style: European
ioi_instrument_option_legs_1_futureRefDate:
ioi_instrument_option_legs_1_delta: 0
ioi_instrument_option_legs_2_strike: 0
ioi_instrument_option_legs_2_expiry:
ioi_instrument_option_legs_2_type:
ioi_instrument_option_legs_2_ratio: 0
ioi_instrument_option_legs_2_underlying_type:
ioi_instrument_option_legs_2_underlying_ticker:
ioi_instrument_option_legs_2_underlying_figi:
ioi_instrument_option_legs_2_exchange:
ioi_instrument_option_legs_2_style:
ioi_instrument_option_legs_2_futureRefDate:
ioi_instrument_option_legs_2_delta: 0
ioi_instrument_option_legs_3_strike: 0
ioi_instrument_option_legs_3_expiry:
ioi_instrument_option_legs_3_type:
ioi_instrument_option_legs_3_ratio: 0
ioi_instrument_option_legs_3_underlying_type:
ioi_instrument_option_legs_3_underlying_ticker:
ioi_instrument_option_legs_3_underlying_figi:
ioi_instrument_option_legs_3_exchange:
ioi_instrument_option_legs_3_style:
ioi_instrument_option_legs_3_futureRefDate:
ioi_instrument_option_legs_3_delta: 0
ioi_instrument_option_structure: CallSpread
ioi_instrument_stock_security_ticker:
ioi_instrument_stock_security_figi:
ioi_goodUntil: 2020-01-23T14:46:21.192+00:00
ioi_bid_price_type: fixed
ioi_bid_price_fixed_price: 83
ioi_bid_price_fixed_currency:
ioi_bid_price_pegged_offsetAmount: 0
ioi_bid_price_pegged_offsetFrom:
ioi_bid_price_pegged_limitPrice: 0
ioi_bid_price_reference:
```

(continues on next page)

(continued from previous page)

```

ioi_bid_price_moneyness: 0
ioi_bid_size_type: quantity
ioi_bid_size_quantity: 1000
ioi_bid_size_quality:
ioi_bid_referencePrice_price: 202
ioi_bid_referencePrice_currency: GBp
ioi_bid_volatility: 0
ioi_bid_notes: bid notes
ioi_bid_qualifiers_count: 0
ioi_bid_qualifiers_0:
ioi_bid_qualifiers_1:
ioi_bid_qualifiers_2:
ioi_bid_qualifiers_3:
ioi_bid_qualifiers_4:
ioi_offer_price_type: fixed
ioi_offer_price_fixed_price: 83
ioi_offer_price_fixed_currency:
ioi_offer_price_pegged_offsetAmount: 0
ioi_offer_price_pegged_offsetFrom:
ioi_offer_price_pegged_limitPrice: 0
ioi_offer_price_reference:
ioi_offer_price_moneyness: 0
ioi_offer_size_type: quantity
ioi_offer_size_quantity: 2000
ioi_offer_size_quality:
ioi_offer_referencePrice_price: 202
ioi_offer_referencePrice_currency: GBp
ioi_offer_volatility: 0
ioi_offer_notes: offer notes
ioi_offer_qualifiers_count: 0
ioi_offer_qualifiers_0:
ioi_offer_qualifiers_1:
ioi_offer_qualifiers_2:
ioi_offer_qualifiers_3:
ioi_offer_qualifiers_4:
ioi_routing_strategy_name:
ioi_routing_strategy_brief:
ioi_routing_strategy_detailed:
ioi_routing_customId:
ioi_routing_broker:
ioi_sentTime:
change:
Ctrl+C pressed. Stopping...
Processing SESSION_STATUS event
SessionConnectionDown = {
    server = "localhost:8194"
}

Session connection is down
Processing SESSION_STATUS event
SessionTerminated = {
}

SessionTerminated = {
}

```

2.6 Error Message

Error Message	Description
Invalid tags in the topic	There is an unsupported tag field or values detected in the topic string.
Exceeded subscription limits	There are too many subscription calls from the client.
The IPER code acronym could not be determined by Bloomberg.	The subscriber's IPER code acronym could not be determined.
The acronym from the IPER code is blank.	The subscriber's acronym code is blank
The side from the IPER code cannot be determined.	The subscriber's side cannot be determined
The UUID is not enabled in NNAB<GO> terminal function.	The subscriber's UUID is not enabled by any broker in NNAB<GO>
Failed to get the user UUID from the subscription call.	There wasn't a valid UUID passed or the UUID passed was set to zero.

2.7 Description of Elements

The following elements are available for equity and options IOI subscription.

The sell-side sending IOIs will buy from the buy-side at the bid size/price and sell to the buy-side at the offer size/price.

Important: All times are in UTC.

Element Name	Description	Type
id_value	Unique Bloomberg value to identify IOI message, also known as handle	string

Continued on next page

Table 1 – continued from previous page

Element Name	Description	Type
ioi_bid_natural	Indicate natural IOI	int32
ioi_offer_natural		
ioi_bid_notes	Notes section for IOI messages	string
ioi_offer_notes		
ioi_bid_price_fixed_currency	Three letter currency acronym for the IOI	string
ioi_offer_price_fixed_currency		
ioi_bid_price_fixed_price	IOI fixed price	float64
ioi_offer_price_fixed_price		
ioi_bid_price_pegged_limit	IOI pegged limit price	float64
ioi_offer_price_pegged_limit	Price	
ioi_bid_price_pegged_offset	IOI pegged offset amount	float64
ioi_offer_price_pegged_offset	Amount	
ioi_bid_price_pegged_offset_from	IOI pegged offset from	string
ioi_offer_price_pegged_offset_from	From	
ioi_bid_price_pegged_offset_type	IOI pegged offset type	string
ioi_offer_price_pegged_offset_type	Type	
ioi_bid_price_reference	Bid, Mid, Ask	string
ioi_offer_price_reference		
ioi_bid_price_type	Market, limit, or unspecified	string
ioi_offer_price_type		
ioi_bid_qualifiers_n	IOI bid/offer qualifiers (e.g. H, U, V, I) [definitions]	string
ioi_offer_qualifiers_n		
ioi_bid_qualifiers_count	IOI bid/offer qualifiers count	int32
ioi_offer_qualifiers_count		
ioi_bid_referencePrice_currency	IOI bid/offer reference currency	string
ioi_offer_referencePrice_currency	Currency	
ioi_bid_referencePrice_price	IOI bid/offer reference price	float64
ioi_offer_referencePrice_price		
ioi_bid_size_quality	Small, Medium, or Large	string
ioi_offer_size_quality		
ioi_bid_size_quantity	Actual quantity of the IOI	int64
ioi_offer_size_quantity		
ioi_bid_size_type	IOI bid/offer size type	string
ioi_offer_size_type		
ioi_bid_volatility	Options IOI bid/offer volatility	float64
ioi_offer_volatility		
ioi_clientId	IOI Client ID	string
ioi_goodUntil	IOI good until time	dateTime
ioi_instrument_option_legs	Options IOI delta	float64
ioi_instrument_option_legs	Options IOI exchange	string
ioi_instrument_option_legs	Options IOI leg expiry	dateTime
ioi_instrument_option_legs	Options IOI future reference date	dateTime
ioi_instrument_option_legs	Options IOI FIGI	string
ioi_instrument_option_legs	Options IOI ticker	string
ioi_instrument_option_legs	Options IOI type	string
ioi_instrument_option_legs	Options IOI ratio	float64
ioi_instrument_option_legs	Options IOI strike	float64
ioi_instrument_option_legs	European, American	string

Continued on next page

Table 1 – continued from previous page

Element Name	Description	Type
ioi_instrument_option_legs	Options-IOI leg type	string
ioi_instrument_option_legs	Options-IOI underlying figi	string
ioi_instrument_option_legs	Options-IOI underlying ticker	string
ioi_instrument_option_legs	Options-IOI underlying type	string
ioi_instrument_option_legs	Options-IOI legs count	string
ioi_instrument_option_stru	Custom, CallSpread, PutSpread, Straddle, Strangle, SingleLegCall, SingleLegPut, CalendarCallSpread, CalendarPutSpread, CallSpreadReversal, PutSpreadReversal, DiagonalCalendarCallSpread, DiagonalCalendarPutSpread, CallButterfly, PutButterfly, IronButterfly, RiskReversal, Box, CallLadder, PutLadder, CallCondor, PutCondor, IronCondor, JellyRoll, RatioCallSpread, RatioPutSpread	string
ioi_instrument_stock_secu	Equity-IOI security figi	string
ioi_instrument_stock_secu	Equity-IOI security ticker	string
ioi_instrument_stock_secu	Equity-IOI security type	string
ioi_instrument_type	IOI instrument type	string
ioi_routing_benchmark		string
ioi_routing_broker		string
ioi_routing_customId		string
ioi_routing_orderType		string
ioi_routing_strategy_brief		string
ioi_routing_strategy_detailed		string
ioi_routing_strategy_name		string
ioi_sentTime	IOI sent time	dateTime
originalId_value		string
state	IOI State: New, Replace and Cancel	string
trader_acronym	IPER code	string
trader_username	Trader name	string
trader_uuid	Trader UUID	int64

2.8 Actionable IOI

The IOIs published via IOI API Publication service can be actionable by the receiving buy-side firms.

The sell-side using IOI API Publication service can specify the targeting EMSX broker code along with `customId` element. The `customId` will allow the order receiving sell-side to tie the order back to the original IOI generated from the sell-side.

Element Name	Description	Type
broker	The broker code used in EMSX to submit the order. This is viewable as <code>ioi_routing_broker</code> element in the IOI API Subscription service.	string
customId	Optional, can be created by the sell-side to correlate back to an order. This is viewable as <code>ioi_routing_id</code> element in the IOI API Subscription service.	string
strategy	Optinal, if specified and the strategy exists in EQMB<GO>, this element will be accepted.	string

3.1 General FAQ

- **What is a session?** Sessions are logical data stream connections and the EMSX API supports failover between physical connections. During this failover, EMSX API will handle re-subscriptions for the end application.

If you are using multiple bloomberg API services, it is recommended to use separate sessions to avoid delaying a fast stream with slow one. For most design, it's best to have separate session for real-time data vs. EMSX API or reference data service.

- **Should I open and close sessions as needed?** No, typically opening and closing a session is expensive for both the client's application and for Bloomberg back-end and thus unnecessary for most application designs while using EMSX API.

CHAPTER 4

Glossary

API Application Programming Interface. The definition of the way in which two applications can communicate with each other.

IOI Indication of Interest (IOI) shows a conditional non-binding sales message from a sell-side or market maker to the buy-side looking to purchase or sell a security.

Bibliography

[definitions] H = Customer Order in Hand - Firm agency order direct from the customer, U = Customer Principal Interest - Firm principal order originating from previous facilitation, V = Switche / Versus Trade, I = In Touch With - , X = For Crossing, W = Working, T = Over Time / Day, D = VWAP, R = Ready to Trade, S = Portfolio Shown.