

## TradingBlox Interface with Interactive Brokers API

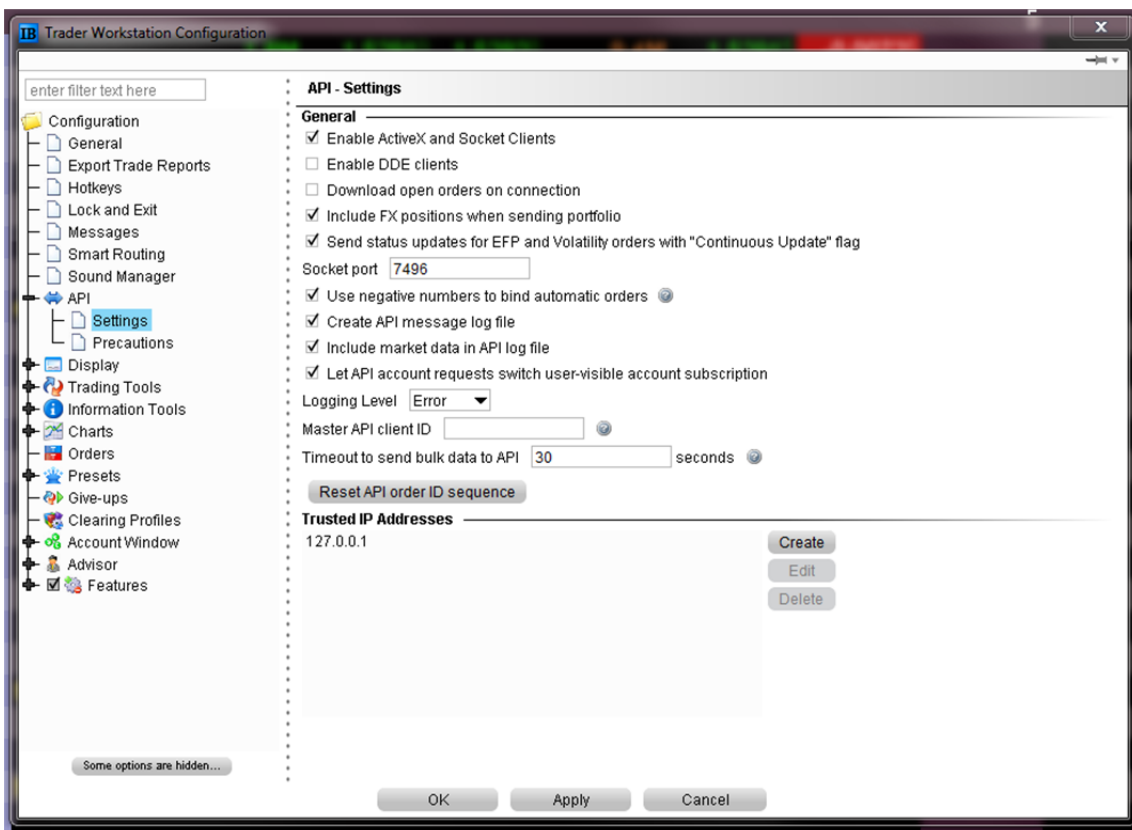
TradingBlox can now be integrated for order submission via the Interactive Brokers API in the Interactive Brokers Traders Workstation (TWS). Using an Auxiliary block with the API command set, you can run TradingBlox Generate Orders and orders will be posted and transmitted through the TWS. Orders must be sent from a TradingBlox Global System Suite.

Please note the following:

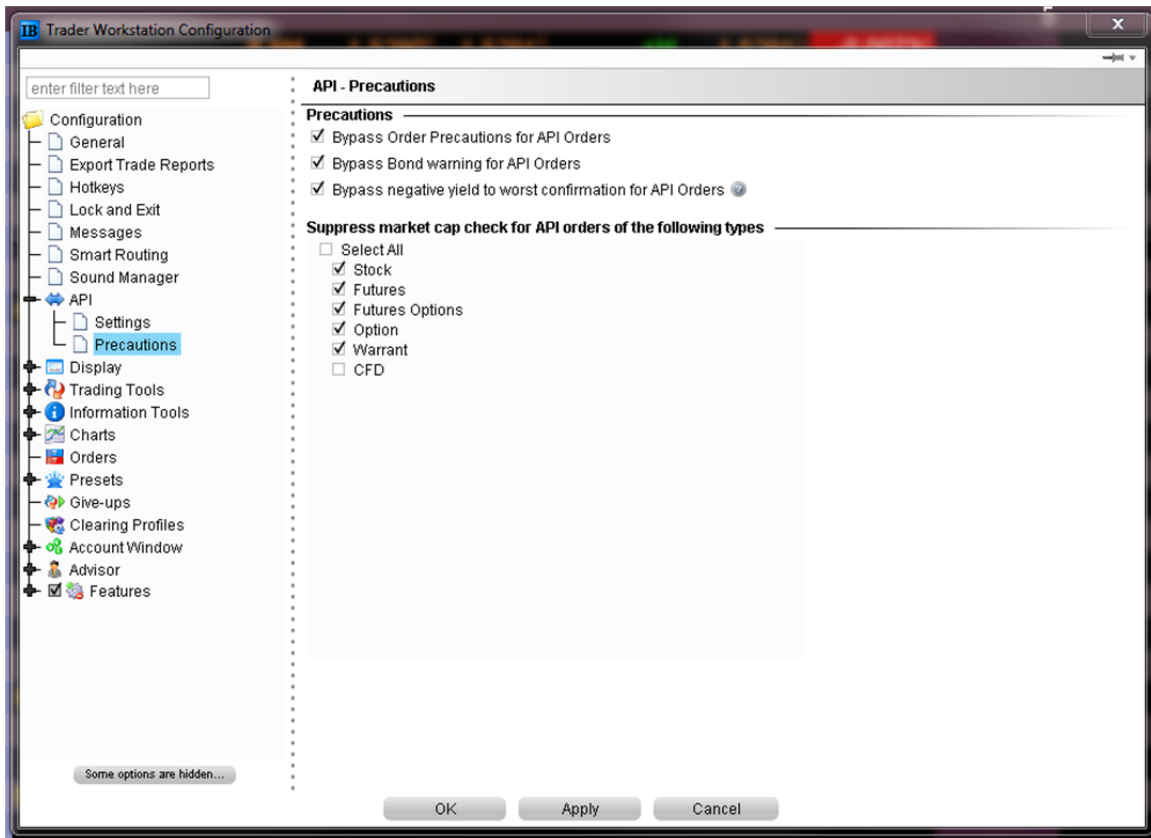
- This process is in Beta mode.
- Current TradingBlox usage is for end of day order generation.
- Users must have the latest version of TradingBlox (4.0.2.0 or later). This capability is not available in earlier versions.
- User License must be enabled for API use.
- TWS must be configured for to utilize the API; see TWS API Configuration section below for settings.
- The Exchange designation for Stocks in the TradingBlox Stock dictionary must be set to SMART.
- Review of market symbols traded is important as IB uses some different symbols than CSI and other data providers. Orders won't get posted if the symbol isn't found in TWS.

### TWS API Configuration Settings

It is highly recommended that users test this functionality in an Interactive Brokers Simulated Trading account before using in an active account. To configure the API for use, go to the Configure button and choose API, Settings. The screen print below shows the recommended settings. The Port number should be 4 digits and unique, but otherwise arbitrary.



Next choose API, Precautions and use the following settings.



## Overview of TWS API Command Set

Here are the IB functions available. Customer License Key needs to be setup to allow IB connection.

The connection is made between Trading Blox and the IB Traders Workstation, which needs to be open when the IBConnect call is made. IBController\* can be used to automate the startup of TWS.

In TWS Global Configurations, Enable Socket Clients, set the Socket port, enter the trusted IP Address, and reset the API order ID sequence when first setting this up.

```
IBConnect( host, port, clientID )
```

The host is typically "" for local host, unless TWS is running on a server.

The port needs to match that in the TWS settings. Trading Blox can connect to multiple instances of TWS if they all have unique port numbers.

The clientID is typically used as a unique system identifier. All orders placed by Trading Blox using this clientID are separate from all others.

### **IBDisconnect ( )**

Disconnects the current open connection between TB and TWS.

### **IBCancelOpenOrders ( )**

Cancels all open orders placed by this client ID and port.

### **system.SetAccountNumber( accountNumber )**

This can be used to set the account number. Used by the sendToIB function to send the order to the correct account, if TWS is handling multiple accounts.

### **order.SendToIB ( )**

Sends the order to IB using the current open connection. The bracket protective stop and profit taking limit prices in the order will automatically be placed as connected orders, unless Entry Day Retracement is set to -1.

Broker Symbol is used, as is the Exchange. These need to be correct for IB, of course.

**orders.SetTimeInForce** and **order.SetClearingIntent** can be used if necessary. These default to "GTC." and "IB".

*\* IBController is a shareware program that adjusts the time in TWS to keep it running during the daily server reset.*