CBOE Application Programming Interface

CBOE API Version 5.3 - Release Notes

Provides an overview of upcoming changes in the next production release of the CMi

CBOE PROPRIETARY INFORMATION

25 November 2008

Document #[API-00]

Front Matter

Disclaimer

Copyright © 1999-2008 by the Chicago Board Options Exchange (CBOE), as an unpublished work. The information contained in this document constitutes confidential and/or trade secret information belonging to CBOE. This document is made available to CBOE members, member firms and other appropriate parties to enable them to develop software applications using the CBOE Market Interface (CMi), and its use is subject to the terms and conditions of a Software License Agreement that governs its use. This document is provided "AS IS" with all faults and without warranty of any kind, either express or implied.

Support and Questions Regarding This Document

Questions regarding this document can be directed to The Chicago Board Options Exchange at 312.786.7300 or via e-mail: api@cboe.com.

The latest version of this document can be found at the CBOE web site: http://systems.cboe.com/webAPI.

Table of Contents

Overview

This document highlights upcoming changes in the new release of the CMi API, Version 5.3. This release supports new IDL constants and documentation changes. The sections below detail the changes in this release. Your feedback or questions regarding this document should be sent to api@cboe.com.

CMi V5.3 Highlights

The highlights discussed below will be available in the December 2008 CBOEdirect software release.

Multiple Linkage Routers

The extensions field in the OrderEntryStruct will be enhanced to allow multiple routing choices for Linkage orders sent through CBOE. The extensions field may not always be present. Valid values for routing are: REDI, OES, BRASS, LIME and ASSENT.

```
module cmiOrder
    struct OrderEntryStruct
      {
        cmiUser::ExchangeFirmStruct executingOrGiveUpFirm;
        string branch;
        long branchSequenceNumber;
        string correspondentFirm;
        string orderDate; // YYYYMMDD format
        cmiUser::ExchangeAcronymStruct originator;
        long originalQuantity;
        cmiProduct::ProductKey productKey;
        cmiUtil::Side side;
        cmiUtil::PriceStruct price;
        cmiOrder::TimeInForce timeInForce;
        cmiUtil::DateTimeStruct expireTime;
        cmiOrder::OrderContingencyStruct contingency;
        cmiUser::ExchangeFirmStruct cmta;
        string extensions;
        string account;
        string subaccount;
        cmiOrder::PositionEffect positionEffect;
```

```
    cmiOrder::CrossingIndicator cross;
    cmiOrder::OriginType orderOriginType;
    cmiOrder::Coverage coverage;
    cmiOrder::NBBOProtectionType orderNBBOProtectionType;
    string optionalData;
    string userAssignedId;
    cmiSession::TradingSessionNameSequence sessionNames;
};
typedef sequence <OrderEntryStruct> OrderEntryStructSequence;
```

New Order Contingency Type

CBOE will support a new order type, Intermarket Sweep Book (ISB) in the W_MAIN session. Orders marked ISB will trade immediately or book. ISB will be represented in CMi constants, const cmiOrder::ContingencyType INTERMARKET_SWEEP_BOOK = 28. Intermarket Sweep Order (ISO) is represented as const cmiOrder::ContingencyType INTERMARKET_SWEEP = 15 and will trade immediately or cancel.

CAS Quote Processing Enhancements

CBOE has redefined its quote processing to allow multiple concurrent quote messages up to a configurable limit. These enhancements will be available in mid December 2008. Please note that, unless explicitly mentioned, all quote messages referred to in the following section are messages acting on the same trading class. A quote method is any of the following requests: acceptQuote, acceptQuotesForClass, cancelQuote, cancelQuotesByClass and cancelAllQuotes.

Due to the asynchronous nature of the concurrent quote message calls; it is very important that the same series not be included in multiple quote blocks. While CBOE will not enforce any restriction with regard to this restriction, the order of processing of the individual calls is *not* guaranteed. The quote processing changes are as follows:

1. The CAS will allow multiple concurrent quote messages up to a configurable limit. These requests include *acceptQuote and acceptQuotesForClass*. The limit at this time is 5.
2. Immediately before the quote message is to be dispatched the quote rate limits are checked. If the call or quote rate limits are not exceeded the message is dispatched.
3. When the number of concurrent quotes in-flight equals the maximum allowed, any new *acceptQuote* or *acceptQuotesForClass* calls will be **rejected** with a NotAcceptedException.

   The error code in this case will be *EXCEEDS_CONCURRENT_QUOTE_LIMIT (4160*) and error text will say, "*Concurrent Quotes exceeded the limit (<limit>)"*.
4. The cancelQuotesByClass and cancelAllQuotes requests will be forwarded to the server immediately without regard to the number of concurrent quote messages currently in progress.

5. While any cancelQuotesByClass is in flight to the server, any new quote request calls (includes *acceptQuote, acceptQuotesForClass and cancelQuote*) will be **rejected** with a NotAcceptedException.

   The error code in this case will be *QUOTE_CANCEL_IN_PROGRESS (4170*) and error text will say, "*Quote Cancel by class is in progress"*.

6. Block quote cancels (block of 0-0 quotes coming through the acceptQuotesForClass calls) are considered to be a acceptQuotesForClass calls and could be rejected if the number of in-flight quote requests to the server exceed the limit or if a cancelQuotesByClass is in-flight.

7. A cancelAllQuotes request is always forwarded to the server. Due to the asynchronous nature of the cancel all quotes request, the CAS will not prevent any new quotes message or quote cancel message (including another cancelAllQuotes) from being forwarded to server while a previous cancel all quotes request is in progress.

Quote Risk Monitor (QRM) Changes

The Quote Risk Monitor (QRM) feature is designed to allow the user to limit the risk associated in quoting. Using the *struct QuoteRiskManagementProfileStruct* in the CMi quote interface, the user can set a threshold limit per class after which the CBOE systems will pull all quotes for the user in the class. Once the threshold the user sets is crossed, the QRM process issues system cancel message for all remaining quotes in the class. The QRM cancel message is out of process from the trading process and the cancel actions are best effort. Meaning it is possible for the user to be filled past the set limit.

The QRM interface has always supported sequences of status messages in the response but prior versions of the QRM feature have only published single status messages. With the software release planned for January 2009, the QRM responses will contain multiple status messages.

IDL Interfaces

New and modified IDL is reflected in **bold** face.

```
module cmiConstants
{
interface ContingencyTypes
  {
    const cmiOrder::ContingencyType NONE = 1; // no contingency
    const cmiOrder::ContingencyType AON = 2; // All or None
    const cmiOrder::ContingencyType FOK = 3; // Fill or Kill
    const cmiOrder::ContingencyType IOC = 4; // Immediate or Cancel
    const cmiOrder::ContingencyType OPG = 5; // Opening only
    const cmiOrder::ContingencyType MIN = 6; // Minimum
    const cmiOrder::ContingencyType NOTHELD = 7; // Not held
    const cmiOrder::ContingencyType WD = 8; // With discretion
```

```
    const cmiOrder::ContingencyType MIT = 9; // Market if touched
    const cmiOrder::ContingencyType STP = 10; // Stop order
    const cmiOrder::ContingencyType STP_LOSS = 11; // Stop loss
    const cmiOrder::ContingencyType CLOSE = 12; // On close
     const cmiOrder::ContingencyType STP_LIMIT = 13; // Stop limit
    const cmiOrder::ContingencyType AUCTION_RESPONSE = 14; // Auction response order
    const cmiOrder::ContingencyType INTERMARKET_SWEEP = 15; // Intermarket sweep
    const cmiOrder::ContingencyType RESERVE = 16; // Reserve order
    const cmiOrder::ContingencyType MIDPOINT_CROSS    = 17; // Mid Point Cross
    const cmiOrder::ContingencyType CROSS             = 18; // Cross
    const cmiOrder::ContingencyType TIED_CROSS        = 19; // Tied cross
    const cmiOrder::ContingencyType AUTOLINK_CROSS    = 20; // Auto link cross
    const cmiOrder::ContingencyType AUTOLINK_CROSS_MATCH = 21; // Auto link cross
    const cmiOrder::ContingencyType CROSS_WITHIN      = 22;
    const cmiOrder::ContingencyType TIED_CROSS_WITHIN = 23;
     const cmiOrder::ContingencyType STOCK_ODD_LOT_NBBO_ONLY = 24;
    const cmiOrder::ContingencyType NBBO_FLASH_THEN_CANCEL = 25;
    const cmiOrder::ContingencyType DO_NOT_ROUTE = 26;
     const cmiOrder::ContingencyType NBBO_FLASH_RESPONSE = 27;
    const cmiOrder::ContingencyType INTERMARKET_SWEEP_BOOK = 28;
};
```

Document Changes

API-01

- No changes

API-02

- Updated the Order Contingency table to include Intermarket_Sweep_Book (ISB).
- Added a new section for “CAS Quote Processing Enhancements” as described above.
- Modified the section: “Recommended Market Making Guidelines” to read as follows:

  It is important to follow the market making guidelines listed below to effectively interact with CBOE’s Hybrid Trading System.

  1. Logout implications. The logout is a relatively expensive function. We have observed Market Makers logging out of the system and immediately logging back in. A less severe method of removing quotes would be to send a ***cancel all quotes*** message.

2. Excessive quoting/thrashing. We have observed firms sending in mass quotes, then single cancels, followed by mass quotes again. If there is a perceived reason validating the need for cancels, then they should not be immediately followed by mass quotes. If the intent is to update the quote, then the cancel is superfluous.
3. There are many single quote blocks entered. It is more efficient to block multiple quotes together within a single message, when possible.
4. Status message handling. A minimum of a single thread per class for status callbacks pertaining to market data is recommended. Status messages should be handled as quickly as possible, in order to allow delivery of the next status message. We ask that each firm use ***class level callbacks for order and quote status***. We have observed that connections cannot keep up with order or quote status if it spans multiple classes with a single callback.

- Modified the QRM section to read as follows:

  The Quote Risk Monitor (QRM) feature is designed to allow the user to limit the risk associated in quoting. Using the *struct QuoteRiskManagementProfileStruct* in the CMi quote interface, the user can set a threshold limit per class after which the CBOE systems will pull all quotes for the user in the class. Once the threshold the user sets is crossed, the QRM process issues system cancel message for all remaining quotes in the class. The QRM cancel message is out of process from the trading process and the cancel actions are best effort. Meaning it is possible for the user to be filled past the set limit.

  QRM is setup at the acronym level. Therefore, multiple users that share the same acronym and exchange will be sharing the same QRM values. For example,

  - User ID1 and user ID2 are sharing acronym ABC
  - User ID1 is quoting IBM
  - User ID2 changes QRM on his/her CBOE*direct* trading system for IBM
  - User ID1's QRM values will change to reflect the changes made by User ID2

  The QRM interface has always supported sequences of status messages in the response but prior versions of the QRM feature have only published single status messages. With the software release planned for January 2009, the QRM responses will contain multiple status messages.

API-03

- Modified the definition of the OrderEntryStruct's extension field to include the values for routing choices on Linkage orders. The values are: REDI, OES, BRASS, LIME and ASSENT.
- Added values for the new constant based on this release
  - const cmiOrder::INTERMARKET_SWEEP_BOOK = 28

API-04

- Modified the definition of the OrderEntryStruct's extension field to include the values for routing choices on Linkage orders. The values are: REDI, OES, BRASS, LIME and ASSENT.
- Added values for the new constant based on this release
    - const cmiOrder::INTERMARKET_SWEEP_BOOK = 28

API-05

- No changes

API-06

- No changes

API-07

- No changes

API-08

- No changes

CAS-01

- No changes

CAS-02

- No changes

Simulator

- No changes

Test Plan Changes

- No changes