



CBOE Application Programming Interface

CBOE API Version 3.2b - Release Notes

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

CBOE PROPRIETARY INFORMATION

06 January 2006

Document #[API-00]

Front Matter

Disclaimer

Copyright © 1999-2006 by the Chicago Board Options Exchange (CBOE), as an unpublished work.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

FRONT MATTER.....	I
DISCLAIMER	I
SUPPORT AND QUESTIONS REGARDING THIS DOCUMENT	I
TABLE OF CONTENTS	2
OVERVIEW	3
CMI V3.2B HIGHLIGHTS	3
SIMPLE AUCTION LIAISON (SAL).....	3
<i>SAL Details</i>	3
CFE OPTIONS ON FUTURES (COF)	4
<i>COF Details</i>	4
CHANGES TO THE CALLBACK REMOVAL ERROR MESSAGE	5
IDL INTERFACES	7
CMICONSTANTS.IDL.....	7
DOCUMENT CHANGES.....	8
API-01	8
API-02	8
API-03	9
API-04	9
API-05	9
API-06	9
API-07	9
CAS-01	9
CAS-02	9
SIMULATOR	9
TEST PLAN CHANGES	9
CMI PHASE 2 TEST PLANS	9

Overview

This document highlights changes for the new release of the CMi API, Version 3.2b. This release details upcoming IDL constant changes to support a new trading session, CFE Options on Futures (COF_MAIN) and a new auction type, Simple Auction Liaison (SAL). In addition, changes to the callback removal error message are detailed below as well as overall documentation changes. Your feedback or questions regarding this document should be sent to api@cboe.com.

CMi V3.2b Highlights

Simple Auction Liaison (SAL)

The auction type, AUCTION_SAL, is introduced in this release. SAL is a mechanism that provides a price-improvement auction for simple (non-complex) orders. Below are the cmicontants that apply to SAL.

```
interface AuctionTypes
{
    const cmicontants::AuctionType AUCTION_SAL = 5;
};
```

The SAL auction type value is 5.

```
interface ActivityReasons
{
    const cmicontants::ActivityReason SAL_IN_PROGRESS = 14;
};
```

This activity reason is used when a SAL auction is in progress.

SAL Details

1. When CBOE is at the NBBO, an eligible marketable order will be stopped at the NBBO and exposed to a brief electronic auction for price improvement. The origins, order size and classes eligible for auction will be configurable.
2. Once the auction begins, quoters who were initially on the NBBO and those that subsequently join or improve the market will not be able to fade or reduce size until the process completes (similar to Quote Trigger).
3. The starting price for the auction will be one penny better than the NBBO.
4. The auction message will be available to all market makers quoting the class (and firms that have orders resting at the top of the market).
5. At the end of the auction period, the order will be executed at the best prices. At each execution price, those that were on the NBBO at the start of the auction will have priority, up to their original size, over those who were not at the NBBO. NBBO participants may also join on quantity in excess of their original size along with all other respondents using the CUMA allocation.

6. DPMs, eDPMs and/or Preferred DPMs will retain a participation right if they were initially on the NBBO and are on the final auction price.
7. An auction will not begin (i.e. the incoming order will simply auto-ex) if the displayed size includes quotes and resting book orders, and if the order size exceeds the quote size.
8. The auction will end early and trade against the existing auction responses and quotes for the following reasons:
 - a. An incoming quote locks or crosses the displayed market.
 - b. An opposite side order is received that is tradable against the SAL order with a quantity equal to or greater than that of the SAL order. If the new order is smaller than the SAL order, it will trade against the SAL order and the auction will continue for the remainder of the SAL order.
 - c. A new marketable order is received on the same side as the SAL order. This order will trade against any remaining responses to the original auction (after the original SAL order is filled), and then a new auction will start for the new order, if appropriate.

CFE Options on Futures (COF)

CFE Options on Futures (COF_MAIN) is a new trading session that allows users to trade options on futures on the CBOE Futures Exchange (CFE). Options on futures is Screen Based Trading only and does not include Hybrid functions. The session will be set to pre-open at 7:58a.m., open at 8:30a.m. and close at 3:02p.m.

COF Details

- Current CFE users who have access to *CBOEDirect* will continue to use the same login and password. The CBOE help desk will set user enablements to allow trading in the COF_MAIN session.
- Supported order contingency types include: Market Order, Limit Order, AON, FOK, IOC, Stop and Stop Limit.
- Market makers entering quotes for options on futures will use the existing `cmiQuote::QuoteEntryStruct`. Users will need to specify the option fields. Required fields include:
 - Underlying Symbol (Futures contract)
 - Series: Month, Strike, Year, Call/Put Code
 - Price
 - Side (Bid, Ask)
 - Quantity
 - Executing Firm
 - Broker Acronym
- Market makers entering orders for options on futures will use the existing `cmiOrder::OrderEntryStruct`. Users will need to specify the option fields. Required fields include:

- Order Type
 - Underlying Symbol (Futures contract)
 - Series:
 - Month
 - Strike
 - Year
 - Call/Put Code
 - Side (Buy, Sell)
 - Price
 - Quantity
 - Account Designation
 - Executing Firm
 - Broker Acronym
 - Time In Force
 - Account Type (Origin)
 - Optional Data Field (if applicable)
- CFE eligible Trading Privilege Holders and Authorized Traders that are not market makers may enter only orders, not quotes, into CBOE*direct*. Users will need to specify the option fields. Required fields include:
 - Order Type
 - Underlying Symbol (Futures contract)
 - Series: Month, Strike, Year, Call/Put
 - Side (Buy, Sell)
 - Price
 - Quantity
 - Time in Force (Day or Good-'til-Canceled)
 - Session
 - Account Type (Origin)
 - Executing Firm
 - Broker Acronym
 - Customer Account Number (Sub-account ID)

Changes to the Callback Removal Error Message

To allow user's to re-register a single removed callback without logging out and resubmitting all callbacks, CBOE has reworked the callback removal error message.

The error message will now contain a standard IIOP reference. This will allow the user to resolve the individual object being removed and reapply only that object instead of all listener objects. Please note – **the IDL did not change**. Only the contents of the fields in the return struct have changed.

The easiest manner in which to test this feature is for the client application to throw an uncaught exception when receiving a callback from the CBOE. Any exception returned to CBOE will force a callback deregistration by the CBOE process.

Existing IDL:

```

cmiCallback.idl:
void acceptCallbackRemoval( in cmiUtil::CallbackInformationStruct
callbackInformation,
                        in string reason,
                        in exceptions::ErrorCode errorCode);
};

cmiUtil:
struct CallbackInformationStruct
{
    string subscriptionInterface;
    string subscriptionOperation;
    string subscriptionValue;
    string ior;
};

```

CBOE used to send the following information in the callback removal:

subscriptionInterface

Callback interface (example: CMICurrentMarketConsumer)

subscriptionOperation

The name of the failed method(acceptCurrentMarket)

subscriptionValue "=" + string representation of object
(com.cboe.idl.CurrentMarketStruct[] @16d58b)

ior

CBOE ORB IOR object digested.

CBOE will now send:

SubscriptionInterface

Callback object type id (example: IDL:cmiCallback/CMICurrentMarketConsumer:1.0)

subscriptionOperation

Method name that failed. Example: acceptCurrentMarket

subscriptionValue

Session class and product key information this object was subscribed for delimited by
"\u0001" and the IOR.

example: =
W_MAIN:69206141^AW_MAIN:69220449^AW_MAIN:89128963^AW_MAIN:69216556

ior

CORBA standard regular ior representation of the CORBA callback object.

IDL Interfaces

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

cmiConstants.idl

```
interface ActivityReasons
{
    const cmiUtil::ActivityReason NOTHING_DONE = 1;
    const cmiUtil::ActivityReason USER = 2;
    const cmiUtil::ActivityReason SYSTEM = 3;
    const cmiUtil::ActivityReason LOST_CONNECTION = 4;
    const cmiUtil::ActivityReason INSUFFICIENT_QUANTITY = 5;
    const cmiUtil::ActivityReason SPECIAL_ADJUSTMENT = 6;
    const cmiUtil::ActivityReason QRM_REMOVED = 7;
    const cmiUtil::ActivityReason INSUFFICIENT_QUANTITY_BUY_SIDE = 8;
    const cmiUtil::ActivityReason INSUFFICIENT_QUANTITY_SELL_SIDE = 9;
    const cmiUtil::ActivityReason QUOTE_UPDATE_CONTROL = 10;
    // acceptServerFailure event would have following reason
    const cmiUtil::ActivityReason FAILOVER = 11;
    const cmiUtil::ActivityReason QUOTE_IN_TRIGGER = 12;
    const cmiUtil::ActivityReason INVALID_SESSION_ID = 13;
    const cmiUtil::ActivityReason SAL_IN_PROGRESS = 14;
};
```

This interface lists the activity reasons in the CMI.

```
interface AuctionTypes
{
    const cmiOrder::AuctionType AUCTION_INTERNALIZATION = 1;
    const cmiOrder::AuctionType AUCTION_STRATEGY = 2;
    const cmiOrder::AuctionType AUCTION_REGULAR_SINGLE = 3; /* not currently used */
    const cmiOrder::AuctionType AUCTION_HAL = 4; /* not currently used */
    const cmiOrder::AuctionType AUCTION_SAL = 5; /* not currently used */
    const cmiOrder::AuctionType AUCTION_UNSPECIFIED = 0; /* not currently used */
};
```

This interface provides the types of auction codes that are supported.

```
interface ProductTypes
{
    const cmiProduct::ProductType COMMODITY = 1;
    const cmiProduct::ProductType DEBT = 2;
    const cmiProduct::ProductType EQUITY = 3;
    const cmiProduct::ProductType FUTURE = 4;
```

```

const cmiProduct::ProductType INDEX = 5;
const cmiProduct::ProductType LINKED_NOTE = 6;
const cmiProduct::ProductType OPTION = 7;
const cmiProduct::ProductType UNIT_INVESTMENT_TRUST = 8;
const cmiProduct::ProductType VOLATILITY_INDEX = 9;
const cmiProduct::ProductType WARRANT = 10;
const cmiProduct::ProductType STRATEGY = 11;
const cmiProduct::ProductType INTEREST_RATE_COMPOSITE = 12;
};

```

Product type values in the CMi.

```

interface Sides
{
    const cmiUtil::Side UNSPECIFIED = 'U';
    const cmiUtil::Side BUY = 'B';
    const cmiUtil::Side SELL = 'S';
    const cmiUtil::Side BID = 'B';
    const cmiUtil::Side ASK = 'A';
    const cmiUtil::Side AS_DEFINED = 'D';
    const cmiUtil::Side OPPOSITE = 'O';
    const cmiUtil::Side SELL_SHORT = 'H';
    const cmiUtil::Side SELL_SHORT_EXEMPT = 'X';
    const cmiUtil::Side BUY_MINUS = 'M';
    const cmiUtil::Side SELL_PLUS = 'P';
};

```

This interface identifies the side of the order.

Document Changes

API-01

- No changes

API-02

- Added a sub-section called: “Callback Removal Error Message” in the section: “Callback Performance Design Issues” to include the callback removal error message changes described in the “Highlights” section of these release notes.
- Included a section for Simple Auction Liaison (SAL) based on this release.
- Updated the Order Contingency Types Table to incorporated COF_MAIN supported order contingency types: Market Order, Limit Order, AON, FOK, IOC, Stop and Stop Limit.
- Added an example for Multi-leg Strategy Orders Entry, Example 14.
 - This code example shows how to request creation of a multi-leg strategy product via CMi. This involves the specification of individual strategy leg products and the quantity and side for each product in the strategy. After creating the strategy, the example code submits an order for the strategy using the acceptStrategyOrder method on the OrderEntry interface. Fields specific to individual strategy legs are included as part of the order entry operation. The usual order status callback method invocations by CBOEdirect for the strategy product are logged to standard output by the example code as the status messages are delivered.

API-03

- Removed the sentence “This method is not allowed for new development” from the CMi quote interface: acceptQuote, cancelQuote and cancelQuotesByClass.
- Defined constants based on this release:
 - `const cmUtil::ActivityReason SAL_IN_PROGRESS = 14`
 - `const cmProduct::ProductType INTEREST_RATE_COMPOSITE = 12`
 - `const cmOrder::AuctionType AUCTION_SAL = 5; /* not currently used*/`

API-04

- Removed the sentence “This method is not allowed for new development” from the CMi quote interface: acceptQuote, cancelQuote and cancelQuotesByClass.
- Defined constants based on this release:
 - `const cmUtil::ActivityReason SAL_IN_PROGRESS = 14`
 - `const cmProduct::ProductType INTEREST_RATE_COMPOSITE = 12`
 - `const cmOrder::AuctionType AUCTION_SAL = 5; /* not currently used*/`

API-05

- No changes

API-06

- No changes

API-07

- No changes

CAS-01

- No changes

CAS-02

- No changes

Simulator

- No changes

Test Plan Changes**CMi Phase 2 Test Plans**

- Section added for verifying heartbeat callbacks are on separate threads.