



---

## **CBOE Application Programming Interface**

### **CBOE API Version 4.1 - Release Notes**

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

## ***CBOE PROPRIETARY INFORMATION***

---

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Copyright © 1999-2006 by the Chicago Board Options Exchange (CBOE), as an unpublished work.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

|   |           |
|---|-----------|
| <b>FRONT MATTER .....</b>                           | <b>I</b>  |
| DISCLAIMER .....                                    | I         |
| SUPPORT AND QUESTIONS REGARDING THIS DOCUMENT ..... | I         |
| <b>TABLE OF CONTENTS .....</b>                      | <b>2</b>  |
| <b>OVERVIEW .....</b>                               | <b>3</b>  |
| <b>CMI V4.1 HIGHLIGHTS.....</b>                     | <b>3</b>  |
| CMI CONSTANTS.....                                  | 3         |
| <b>IDL INTERFACES .....</b>                         | <b>4</b>  |
| CMICONSTANTS.IDL.....                               | 5         |
| <b>DOCUMENT CHANGES.....</b>                        | <b>8</b>  |
| API-01 .....  | 8         |
| API-02 .....  | 8         |
| API-03 .....  | 9         |
| API-04 .....  | 9         |
| API-05 .....  | 9         |
| API-06 .....  | 9         |
| API-07 .....  | 9         |
| API-08 .....  | 9         |
| CAS-01 .....  | 9         |
| CAS-02 .....  | 9         |
| <b>SIMULATOR .....</b>                              | <b>10</b> |
| <b>TEST PLAN CHANGES .....</b>                      | <b>10</b> |

## Overview

Release notes for the CMi API, Version 4.1, were initially published on September 8, 2006. Version 4.1 supports new order types for CBOE's Stock system. Since the initial publication, new "Activity Reasons" constants have been defined. IDL and documentation changes are detailed in the sections below. No simulator changes are required. Your feedback or questions regarding this document should be sent to [api@cboe.com](mailto:api@cboe.com).

## CMi V4.1 Highlights

### CMi Constants

#### Order Contingency Types

Several new order contingency types are introduced in this release for the CBOE Stock system. Stock trading on CBOEdirect supports new order contingency types: Intermarket Sweep orders (ISO), Reserve orders and Cross orders. Firms use the OrderEntry.acceptOrder interface to transmit ISO and Reserve orders.

Firms can send Cross orders as a pair using the OrderEntry.acceptCrossingOrder() interface or as two single orders via the OrderEntry.acceptOrder interface. Using the OrderEntry.acceptCrossingOrder enables the user to send two orders (BUY and SELL) in one message.

#### (1) **const cmOrder::ContingencyType INTERMARKET\_SWEEP = 15; // Intermarket sweep order**

Intermarket Sweep Orders (ISO) will be implemented in the RegNMS release of the Stock system, which is scheduled for October 2006. Intermarket Sweep Orders are treated as IOC orders but will trade against the book without regard to the NBBO. Trades executed as a result of these orders are exempted from RegNMS trade-through rules. ISO orders may only be sent if the sender has simultaneously sent orders to any other markets with protected quotes priced better than the limit price on the ISO.

#### (2) **const cmOrder::ContingencyType RESERVE = 16; // Reserve order**

Reserve orders are a new feature added in Stock Hybrid, which is scheduled for implementation in January 2007. A reserve order has two quantities associated with it, order quantity and display quantity. Only the display quantity is visible as CBOE's book. The remaining quantity (order quantity - display quantity) is available to trade but not visible.

In addition, new cross orders have been added to the Stock Hybrid system. In all cases, the matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled. Below is a brief description of each cross type.

#### (3) **const cmOrder::ContingencyType MIDPOINT\_CROSS = 17; // Mid Point Cross**

Mid Point cross: These two orders can arrive as a pair (together) or one after the other. In either case they will need the same contingency. A mid-point-cross trades at the middle

of current NBBO and will trade at ½ cent increments. Both orders must be for the same size.

**(4) const cmiOrder::ContingencyType CROSS = 18; // Cross**

These two orders can arrive as a pair (together) or one after the other. In either case they will need the same contingency. If they are for the same price and size they will trade against each other immediately as long as the price is at or within CBOE's quote and the NBBO. If the trade price is equal to the CBOE's best market and the market includes public customer orders, the cross order must be: X shares or more; be for a dollar amount greater than or equal to \$Y; and larger than any public customer interest at that price. Both orders must be for the same price and size and neither order will execute unless both orders are received.

**(5) const cmiOrder::ContingencyType TIED\_CROSS = 19; // Tied cross (CURRENTLY UNSUPPORTED)**

Tied Cross: Similar to Cross orders except that they can trade even if they are same as the CBOE's current market. That is the trade price does not have to be better than CBOE's current market.

**(6) const cmiOrder::ContingencyType AUTOLINK\_CROSS = 20; // Auto link cross**

Auto Link Cross is an order that will be Autolinked if CBOE is not the NBBO and the order is tradable at other markets. If there is remaining quantity, then this order will trade against an AutoLink\_Cross\_Match order (see below). Autolink\_cross orders and autolink\_cross\_match orders can route in a single paired message or as two separate orders. Both orders must be for the same price but do not necessarily need to be for the same size. The autolink\_cross order will not execute unless an autolink\_cross\_match order is received.

**(7) const cmiOrder::ContingencyType AUTOLINK\_CROSS\_MATCH = 21; // Auto link cross**

Users have to submit two orders for AUTOLINK crosses. One is AUTOLINK\_CROSS and the second one is AUTOLINK\_CROSS\_MATCH. If away markets are better, then AUTOLINK\_CROSS will first sweep away markets and the CBOE book. If there is still quantity remaining, then it will trade against the AUTOLINK\_CROSS\_MATCH.

## Activity Reasons

New Activity Reasons have been defined to indicate why an order was rejected/cancelled by CBOE.

**(1) const cmiUtil::ActivityReason CROSS\_IN\_PROGRESS = 15;**

The order was the same as the resting Cross order. The resting Cross order did not route as a pair and is waiting to trade.

**(2) const cmiUtil::ActivityReason INVALID\_NBBO = 16;**

Signifies that the NBBO was invalid for the order.

**(3) const cmiUtil::ActivityReason NOT\_WITHIN\_NBBO = 17;**

The Cross order price was not within the NBBO.

**(4) const cmiUtil::ActivityReason TRADE\_THROUGH\_CBOE = 18;**

The order price traded through CBOE's market.

**(5) const cmiUtil::ActivityReason INSUFFICIENT\_CUSTOMER\_ORDER\_QUANTITY = 19;**

The Cross order quantity was less than the customer order quantity at CBOE's best market.

**(6) const cmiUtil::ActivityReason INSUFFICIENT\_CROSS\_ORDER\_SIZE = 20;**

The Cross order size was less than the defined minimum order size.

**(7) const cmiUtil::ActivityReason  
INSUFFICIENT\_CROSS\_ORDER\_DOLLAR\_AMOUNT = 21;**

The Cross order price was less than the defined minimum dollar amount.

**(8) const cmiUtil::ActivityReason SELL\_SHORT\_RULE\_VIOLATION = 22;**

The order violated the Sell Short rule.

## IDL Interfaces

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

### **cmiConstants.idl**

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
}

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;

    // The following are used for Stock
    const cmiUtil::ActivityReason CROSS_IN_PROGRESS = 15;
    const cmiUtil::ActivityReason INVALID_NBBO = 16;
    const cmiUtil::ActivityReason NOT_WITHIN_NBBO = 17;
    const cmiUtil::ActivityReason TRADE_THROUGH_CBOE = 18;
    const cmiUtil::ActivityReason INSUFFICIENT_CUSTOMER_ORDER_QUANTITY = 19;
    const cmiUtil::ActivityReason INSUFFICIENT_CROSS_ORDER_SIZE = 20;
    const cmiUtil::ActivityReason INSUFFICIENT_CROSS_ORDER_DOLLAR_AMOUNT = 21;

```



**const cmiUtil::ActivityReason SELL\_SHORT\_RULE\_VIOLATION = 22;**

// The following are used for Linkage

```
const cmiUtil::ActivityReason BROKER_OPTION = 100;
const cmiUtil::ActivityReason CANCEL_PENDING = 101;
const cmiUtil::ActivityReason CROWD_TRADE = 102;
const cmiUtil::ActivityReason DUPLICATE_ORDER = 103;
const cmiUtil::ActivityReason EXCHANGE_CLOSED = 104;
const cmiUtil::ActivityReason GATE_VIOLATION = 105;
const cmiUtil::ActivityReason INVALID_ACCOUNT = 106;
const cmiUtil::ActivityReason INVALID_AUTOEX_VALUE = 107;
const cmiUtil::ActivityReason INVALID_CMTA = 108;
const cmiUtil::ActivityReason INVALID_FIRM = 109;
const cmiUtil::ActivityReason INVALID_ORIGIN_TYPE = 110;
const cmiUtil::ActivityReason INVALID_POSITION_EFFECT = 111;
const cmiUtil::ActivityReason INVALID_PRICE = 112;
const cmiUtil::ActivityReason INVALID_PRODUCT = 113;
const cmiUtil::ActivityReason INVALID_PRODUCT_TYPE = 114;
const cmiUtil::ActivityReason INVALID_QUANTITY = 115;
const cmiUtil::ActivityReason INVALID_SIDE = 116;
const cmiUtil::ActivityReason INVALID_SUBACCOUNT = 117;
const cmiUtil::ActivityReason INVALID_TIME_IN_FORCE = 118;
const cmiUtil::ActivityReason INVALID_USER = 119;
const cmiUtil::ActivityReason LATE_PRINT = 120;
const cmiUtil::ActivityReason NOT_FIRM = 121;
const cmiUtil::ActivityReason MISSING_EXEC_INFO = 122;
const cmiUtil::ActivityReason NO_MATCHING_ORDER = 123;
const cmiUtil::ActivityReason NON_BLOCK_TRADE = 124;
const cmiUtil::ActivityReason NOT_NBBO = 125;
const cmiUtil::ActivityReason COMM_DELAYS = 126;
const cmiUtil::ActivityReason ORIGINAL_ORDER_REJECTED = 127;
const cmiUtil::ActivityReason OTHER = 128;
const cmiUtil::ActivityReason PROCESSING_PROBLEMS = 129;
const cmiUtil::ActivityReason PRODUCT_HALTED = 130;
const cmiUtil::ActivityReason PRODUCT_IN_ROTATION = 131;
const cmiUtil::ActivityReason STALE_EXECUTION = 132;
```

```
const cmiUtil::ActivityReason STALE_ORDER = 133;
const cmiUtil::ActivityReason ORDER_TOO_LATE = 134;
const cmiUtil::ActivityReason TRADE_BUSTED = 135;
const cmiUtil::ActivityReason TRADE_REJECTED = 136;
const cmiUtil::ActivityReason ORDER_TIMEOUT = 141;
const cmiUtil::ActivityReason REJECTED_LINKAGE_TRADE = 170;
const cmiUtil::ActivityReason SATISFACTION_ORD_REJ_OTHER = 171;
const cmiUtil::ActivityReason PRODUCT_SUSPENDED = 172;

// Currently used for TPF linkage; in future may be used for CBOEdirect
const cmiUtil::ActivityReason UNKNOWN_ORDER = 137;
const cmiUtil::ActivityReason INVALID_EXCHANGE = 138;
const cmiUtil::ActivityReason TRANSACTION_FAILED = 139;
const cmiUtil::ActivityReason NOT_ACCEPTED = 140;

// Used for linkage when cancel reason is not provided (could be user cancel or cancel remaining)
const cmiUtil::ActivityReason AWAY_EXCHANGE_CANCEL = 199;

// Linkage Business Message Reject codes
const cmiUtil::ActivityReason LINKAGE_CONDITIONAL_FIELD_MISSING = 900;
const cmiUtil::ActivityReason LINKAGE_EXCHANGE_UNAVAILABLE = 901;
const cmiUtil::ActivityReason LINKAGE_INVALID_MESSAGE = 902;
const cmiUtil::ActivityReason LINKAGE_INVALID_DESTINATION = 903;
const cmiUtil::ActivityReason LINKAGE_INVALID_PRODUCT = 904;
const cmiUtil::ActivityReason LINKAGE_SESSION_REJECT = 905;
};
```

## Document Changes

### API-01

- No changes

### API-02

- Added a new section “Support for Stock Order Types”.
- Updated the “Order Contingency Types” table to include the contingency types in this release.

- Changed “Preferred DPM” to read “Preferred Market Maker” throughout the document.
- Defined the Expected Opening Price role as:
  - Main purpose is to display expected opening price and expected opening size messages
  - User ID must be created and assigned by the CBOE Help Desk
  - Not allowed to enter orders
  - Not allowed to enter quotes
  - Not allowed to send RFQs and not allowed to receive RFQs
  - Has full access to viewing products, market data, strategies
  - Not able to create strategy products

**API-03**

- Described new messages and data types based on this release.

**API-04**

- Defined new constants based on this release.

**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