CBOE Application Programming Interface

Version 9.0.2

CBOE API Volume 1: Overview and Concepts

Provides an overview of the CBOE CMi application programming interface and presents concepts important in understanding and using the CBOE APIs.

CBOE PROPRIETARY INFORMATION

15 July 2011

Document #[API-01]

Front Matter

Disclaimer

This document is provided “AS IS” with all faults and without warranty of any kind, either express or implied.

Table of Contents

Change Notices

The following change notices are provided to assist users of the CMi in determining the impact of changes to their applications.

Date	Version	Description of Change
15 Jul 2011	V9.0.2	Removed references to COMPASS
29 Apr 2011	V9.0.1	No changes
14 Jan 2011	V9.0	No changes
08 Jan 2010	V7.0	No changes
12 Aug 2009	V6.1	No changes
22 May 2009	V6.0	No changes
25 Nov 2008	V5.3	No changes
24 Sept 2008	V5.2	No changes
23 Jul 2008	V5.1	No changes
29 Feb 2008	V5.0	No changes
18 Jan 2008	V4.2.4	No changes
26 Oct 2007	V4.2.3	No changes
01 June 2007	V4.2.2	No changes
23 Feb 2007	V4.2.1	No changes
15 Dec 2006	V4.2	No changes
20 Sept 06	V4.1	No changes
25 May 06	V4.0	Referenced the new MDX document, API-08
06 Jan 2006	V3.2b	No updates
12 Aug 2005	V3.2	No updates
29 Jul 2005	V3.2	No changes
08 Apr 2005	V3.1a	No changes
30 Nov 2004	V3.1	No changes
18 Jun 2004	V3.0	No changes.
28 Apr 2004	V2.52	No changes.
06 Feb 2004	V2.63	No changes.
10 Oct 2003	V2.62	No changes.
29 Aug 2003	V2.61	Method description changes.
13 Jul 2003	V2.6	Support for Market Linkage and Stock.

Date	Version	Description of Change
08 Jul 2003	V2.51	Revisions since the last release.
14 Mar 2003	V2.5	Support for Hybrid
24 Jan 2003	V2.1	Support for Linkage P orders.
20 May 2002	V2.0.1	No change.
22 Apr 2002	V2.0	Production Release
27 Feb 2002	V2.0b	Software Development Kit Beta 2
23 Jan 2002	V2.0a	Software Development Kit Beta 1
14 Dec 2001	V2.0	Updated documentation only for Version 2.0.
27 Apr 2001	V1.0b	Added Market Data role information.
16 Mar 2001	V1.0a	Error corrections and updated to reflect that Strategies will not be part of Version 1.0.
15 Jan 2001	V1.0	Production Version.
15 Sep 2000	V0.9	Network testable version.
28 Apr 2000	V0.8	Includes revisions to the CMi API since the last update. Refer to the Release Notes for full details.
30 Sep 1999	V0.5	Modifications to the use of FIX protocol. Changes to some of the questions in the API.
9 July 1999	V0.4	Updated FIX Protocol support section—added questions section. Minor copy editing and drawing changes.

About This Document

Purpose

This document provides an overview of the CBOE Application Programming Interfaces (APIs). Concepts that are necessary to understand and use the CBOE APIs are also presented.

Intended Audience

Anyone who has a need to use or support CBOE application programming interfaces.

Related Documents

Document Number	Document Description
Roadmap.doc	CBOE API and CAS Document Road Map
API-02	CBOE API Volume 2: CMi Programmer's Guide to Interfaces and Operations
API-03	CBOE API Volume 3: CMi Programmer's Guide to Messages and Data Types
API-04	CBOE API Volume 4: CMi Dictionary of Attributes and Operations
API-05	CBOE API Volume 5: Using CMi with Specific Object Request Brokers
API-06	CBOE API Volume 6: Connecting to the CBOE Network
API-07	CBOE API Volume 7: CBOEdirect Certification and Testing Procedures
API-08	CBOE API Volume 8: CMi Programmer’s Guide to the Market Data Express (MDX) Data Feed
CAS-01	CBOE Application Server Volume 1: Overview and Concepts
CAS-02	CBOE Application Server Volume 2: CBOE Application Server Simulator for Stand Alone Testing
FIX-01	CBOE API FIX Protocol Support Volume 1: Overview

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.

Introduction

The CBOE Market Interface (CMi) is a distributed object interface based upon the CORBA (Common Object Request Broker Architecture) standard from the Object Management Group (OMG). The interface is defined using the Interface Definition Language (IDL), which is an OMG and ISO standard. Messages are transported using the Internet Inter-Orb Protocol (IIOP), which operates over standard Internet protocols (TCP/IP).

The CMi is one of the two interfaces available to the CBOEdirect system, the others being the FIX 4.2 service.

This document presents an overview of the CBOE Market Interface (CMi).

System Overview

The combination of CBOEdirect and the existing systems that support the open outcry markets are referred to collectively in this document as the *CBOE Exchange Services*. One of the primary design goals of the API is to permit members to develop applications that are independent of the location and style of trading for a given CBOE product.

To understand CBOE's approach, it is important to understand some basic requirements and goals of the API initiative.

Impetus for CMi

The CMi was designed and built to adhere to the following requirements:

- Provide a common interface for securities trading, whether that trading is occurring in the open outcry market or via an electronic trading system.
- Be based upon industry standards where possible.
- Be relatively simple to use.
- Provide high performance access.
- Provide high levels of security and message integrity.
- Be compatible with Internet protocols to exploit the ubiquity and relative low cost of Internet technologies.

CBOEdirect System Architecture

CBOE is migrating its applications and adding screen based trading using a multi-tier architecture. The multi-tier architecture has proven to provide scalability to meet the high volume requirements inherent in CBOE's markets.

Tier	Architecture Component	Responsibility
1	CBOE Workstation, Member Application Program or third	Presentation and interface to exchange services

Tier	Architecture Component	Responsibility
	party GUI	
2	CBOE Client Application Server	Session management, authentication of user, caching of information to reduce bandwidth
3	Fault tolerant Message Routing and Distribution	Message bundling, routing and distribution, reliable message delivery
4	CBOEdirect Front End Processors	Application level routing between clients and exchange services
5	CBOEdirect Services	Provides exchange services, such as trading, market data dissemination

This architecture also provides for access to services that support both electronic and open outcry trading.

CBOE Workstation (Tier 1)

The CBOE Workstation is the name given to a set of interactive, graphical user interface (GUI) applications provided by CBOE that support member trading needs

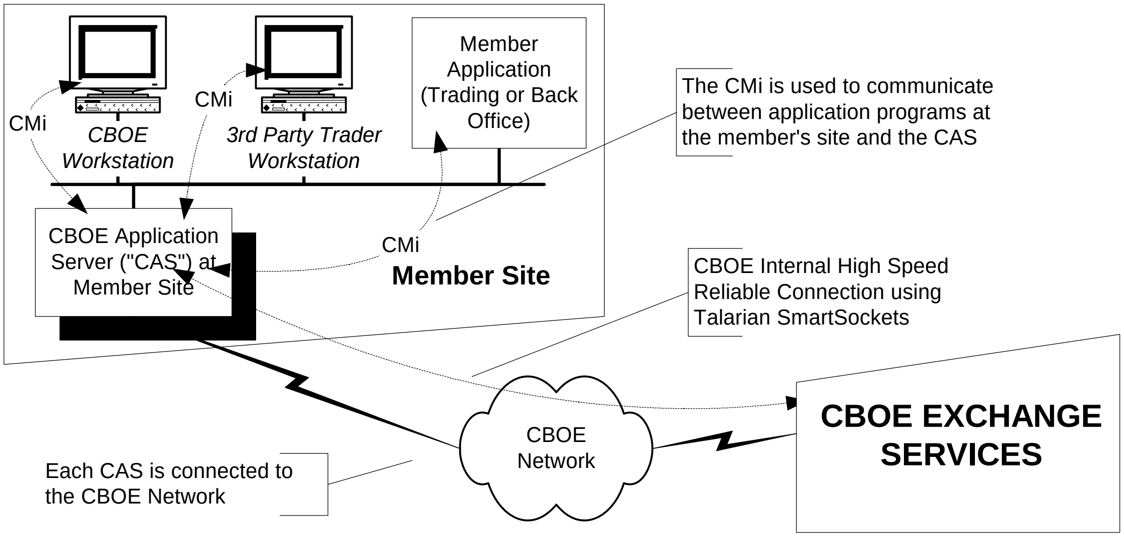
Member or Third Party Programs (Tier 1)

Members can choose to build their own applications to operate using the CMi. CBOE will make every attempt to fully support the membership in creating their custom or third party applications.

CBOE Application Server (CAS) (Tier 2)

The CBOE Application Server (CAS) is a platform independent, highly portable, robust application, written in Java that acts as a proxy and smart cache between the client application program and the exchange services. More precisely, the CAS sits between a client application, such as the CBOE Workstation or your own custom trading application, and the *CBOEdirect Front End Processors*.

By using an Application Server approach, CBOE is able to minimize bandwidth consumption by minimizing the amount of duplicate information going to multiple clients. A theme throughout the CBOE Architecture is the use of caching to minimize bandwidth utilization.


Member Application (Trading or Back Office)
CMi
CMi
CBOE Workstation
3rd Party Trader Workstation
The CMi is used to communicate between application programs at the member's site and the CAS
CBOE Application Server ("CAS") at Member Site
CMi
Member Site
CBOE Internal High Speed Reliable Connection using Talarian SmartSockets
CBOE EXCHANGE SERVICES
CBOE Network
Each CAS is connected to the CBOE Network

Each CAS is connected to two CBOEdirect Front End Processors via the CBOE Network.

CBOE Network and Messaging Infrastructure (Tier 3)

CBOE designed and implemented a fault-tolerant, high performance infrastructure using a combination of third party software, industry standards, and custom in-house development software.

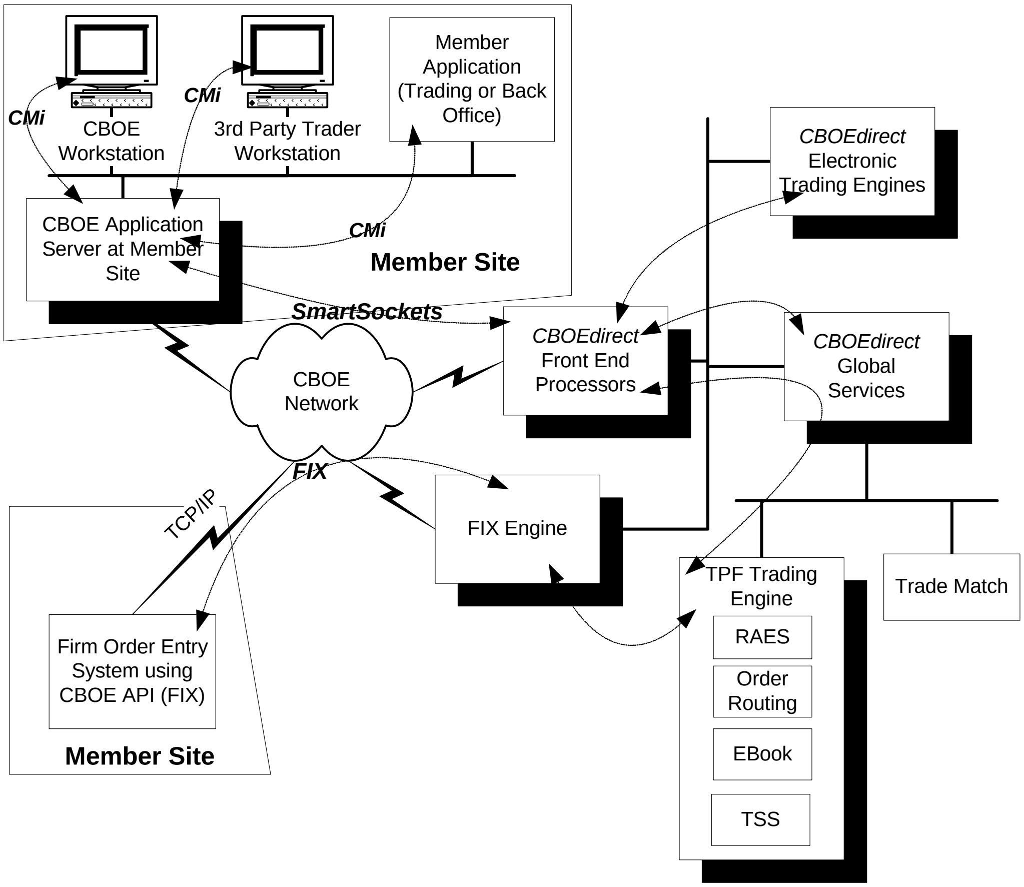
CBOEdirect Front End Processor (FEP) (Tier 4)

The CBOEdirect Front End Processor provides for application level routing between the CBOE Application Servers and the CBOE Exchange Services. The FEP also provides a primary and secondary access to CBOE Exchange Services.

CBOE Exchange Services (Tier 5)

The services support electronic and open outcry markets. The Exchange Services include:

- Order Routing and Handling to both CBOEdirect and the Order Routing System (ORS) that supports open outcry markets.
- Market Data Publication
- CBOEdirect Electronic Trading Engines.
- Trade reporting


CMi
CBOE Workstation
CMi
3rd Party Trader Workstation
Member Application (Trading or Back Office)
CBOE Application Server at Member Site
CMi
Member Site
SmartSockets
CBOE Network
CBOEdirect Electronic Trading Engines
CBOEdirect Front End Processors
CBOEdirect Global Services
FIX
TCP/IP
FIX Engine
TPF Trading Engine
RAES
Order Routing
EBook
TSS
Trade Match
Firm Order Entry System using CBOE API (FIX)
Member Site

Implementing An Application Using The CMi

Development

Organizations wishing to interface to CBOE exchange services will first start by signing the License Agreement and then downloading the CBOE CMi Software Development Kit ("SDK") from the CBOE API web site. The SDK contains software, plus current documentation and example code that can be used as a starting point for development.

Testing and Certification

Firms may not participate in live CBOEdirect markets without completing testing and becoming certified. The testing and certification of your application is done in phases. Each testing phase must be performed in sequential order. Each phase is described in detail in document API-07 (CMi Volume 7) which is available for download at the CBOE API web site.

CBOE Application Programming Interfaces

CBOE has chosen to support two different interfaces to access screen-based trading services. The first is a robust, full function API using the CORBA/IIOP protocol, which is referred to as the CBOE Market Interface (CMi). This version of the API is intended for market makers and traders that plan on using high bandwidth functions, such as market data, and in the case of market makers, auto-quotation facilities.

Retail firms and institutional customers will likely be interested in adopting the second new interface, CBOE's implementation of FIX 4.2. API FIX Protocol. CBOE is providing this protocol based upon the wide acceptance of FIX as a standard for order entry, request for quotation (indication of interest), and execution reporting. CBOE is participating in expanding the applicability of FIX to exchange based trading.

Depending on existing systems, some firms may choose to adopt both APIs for different applications.

CBOE Market Interface (CMi)

CBOE has selected CORBA/IIOP as the application programming interface between the CBOE Application Server and client application programs, such as the CBOE Workstation.

CORBA/IIOP is part of the Object Management Architecture (OMA) that is defined by the Object Management Group (OMG) a consortium of over 500 firms with interest in open standards for distributed computing.

At the core of OMA is a language known as the Interface Definition Language, which is now an international standard. The IDL defines the interface between the client and the server in terms of object oriented description similar to C++ or Java class declarations.

Questions about CORBA/IIOP

What is IIOP?

IIOP is the specification of message formats for communicating to distributed objects over a TCP/IP (or Internet) network. The Internet Inter-Orb Protocol (IIOP) has been specified by the OMG to provide interoperability between ORBs supplied by different companies. Companies supporting IIOP provide for a high degree of Interoperability. Meaning each participant in a distributed computing environment can select the ORB product that suits their own business needs and use it to communicate to another program written using a different ORB product. As with any standard, there are rules that must be followed for this network "nirvana" to be achieved. CBOE conducts testing to verify interoperability of its application with different vendors products. At this time, CBOE supports the following products:

- VisiBroker from Inprise for Java 5.x
- Microsoft Visual C++
- Sun JDK 1.4.2
- Javasoft JDK 1.4 ORB provided with the Java Development Kit.

Other ORB products can be validated, assuming enough interest from members.

Why did CBOE select CORBA/IIOP?

CORBA/IIOP is a standard for distributed computing that uses an object-oriented paradigm. Instead of recreating the software libraries for communication between client applications and the exchange, adoption of the CORBA standard means that CBOE and its members have access to commercial off the shelf software and also open source software packages.

CORBA/IIOP is an official Internet protocol and is supported by several companies.

CORBA/IIOP provides for the separation of interface and implementation - meaning that the interface between the system is rigorously and exactly defined in the CORBA Interface Definition Language (IDL). The CORBA IDL is similar to class declarations in C++ or Java. The CORBA IDL serves as the contract between the client and the server. The IDL defines the interface - it leaves the implementation up to the user. The implementation is created using a commercial Object Request Broker (ORB). The ORB is the software that sits between an application program accessing an object and the implementation of the object.

CORBA/IIOP supports multiple programming languages. The OMG has adopted mapping from OMG IDL to Java, C, C++, SmallTalk, Ada, COBOL. CBOE provides examples in Java and C++.

There are CORBA implementations for all major hardware and operating system platforms.

CORBA has a set of services that usually are custom built by developers of distributed applications. The CORBA services are specifications of standard functionality needed in distributed systems, e.g. Name Service, Event Service, Time Service. CORBA/IIOP would allow the direct use of those services.

CORBA/IIOP has wide support. There are currently about twenty ORB products available.

CORBA/IIOP is an excellent facility for integrating legacy applications (such as the CBOE TPF Trading Engine).

Doesn’t CORBA/IIOP and distributed objects in general cause performance problems?

Naïve use of CORBA to create a distributed object based application can cause significant negative performance problems. The idea of distributing an application across a network, especially a wide area network, can lead to both reliability and scalability problems.

CBOE uses CORBA as a transport layer for the transmission of messages to and from business objects. An examination of the CBOE CMi IDL will reveal that CBOE has been extremely careful in designing the IDL to minimize network bandwidth consumption, so as to eliminate potential performance problems.

Microsoft does not support CORBA, what impact will this have if we have standardized on Microsoft platforms and architectures?

Although Microsoft does not directly support CORBA, the OMG has created a specification for interoperability between Microsoft's proprietary distributed object model, the Distributed Component Object Model (DCOM) and CORBA. Several vendors have excellent products that provide access to CORBA objects through Microsoft's DCOM. One CORBA vendor has even gone so far as to license the source code to DCOM to provide a very high degree of compatibility and interoperability.

In addition, almost all CORBA vendors have robust implementations of their ORB products that operate on the Microsoft Windows NT operating system.

Glossary

Term	Abbreviation	Definition
Application Programming Interface	API	A set of routines, protocols, and tools for building software applications.
CBOE Application Server	CAS	Computing tier that provides caching, session management, system access to applications requiring access to Exchange Services. Clients access the CBOE Application Server via the CBOE Market Interface or the CBOE FIX interface.
CBOE Application Server Simulator		Standalone version of the CAS to be used by developers and for standalone testing.
CBOEdirect		The new CBOE electronic trading system
CBOE Front End Processor		Computing tier responsible for application level routing between Application Servers and Exchange Services.
CBOE Exchange Services		Services provided by an exchange, such as order entry and routing, order matching, reporting, and market data services.
CBOE Market Interface	CMi	The CORBA based interface from the CBOE that provides access to CBOEdirect and to CBOE open outcry markets.
CBOE Network		The private managed network provided by CBOE to access CBOEdirect and other CBOE exchange services.
CMi Client		User program that uses the CMi API to access the CBOE.
COMPASS		CBOE's order entry and routing API.
Common Object Request Broker Architecture	CORBA	Standard from the Object Management Group for communication to/from objects in a heterogeneous network environment.
Ethernet		A local-area network (LAN) protocol.
Extranet		An intranet that is partially accessible to authorized outsiders. Whereas an intranet resides behind a firewall and is accessible only to people who are members of the same company or organization, an extranet provides various levels of accessibility to outsiders. You can access an extranet only if you have a valid username and password, and your identity determines which parts of the extranet you can view.
Firewall		An electronic boundary that prevents unauthorized

Term	Abbreviation	Definition
		access to a network.
Financial Information Exchange Protocol	FIX	Open standard conceived of by a group of institutions and brokers interested in streamlining the trading process. It is now used by a variety of firms and vendors.
Gateway		A computing device, such as a router, computer, or firewall, that sits between two networks to deliver messages between the two networks.
Interface Definition Language	IDL	The language specified by the Object Management Group to describe object interfaces in CORBA
Internet Inter-Orb Protocol	IIOP	Protocol that maps a collection of CORBA message requests to TCP/IP.
Interoperable Object Reference	IOR	Portable method for referencing CORBA objects
Internet Protocol	IP	Protocol that is designed to be independent of underlying physical network structure. Uses an independent numbering scheme (IP Addressing) to provide network messaging that can be independent and easily routed between heterogeneous networks. Part of the TCP/IP Protocol Suite.
International Organization for Standardization	ISO	International Organization for Standarization
Local Area Network	LAN	Local Area Network
Network Address Translator	NAT	Network Address Translator – The conversion of the source and/or destination IP address on a packet when traversing a gateway, typically a firewall.
Object Management Architecture	OMA	The architecture that defines how objects can interoperate in a heterogeneous computing environment, as defined by the Object Management Group.
Object Management Group	OMG	Standards organization consisting of 500+ member organizations, primarily software and hardware vendors. The OMG is responsible for maintaining the Object Management Architecture (OMA), which includes the CORBA standards.
Object Request Broker	ORB	Software that is responsible for managing distributed objects in a computing environment. The ORB is responsible for resolving locations of objects and establishing communication between clients and object services.
Open Shortest Path First		Open Shortest Path First – An IP interior Routing Protocol

Term	Abbreviation	Definition
Market Data Express	MDX	CBOE's market data feed that supports the reporting of quotes, trades, and other information for options, stocks and futures.
Point to Point Internet Protocol	PPP	IP protocol optimized for connection over point to point dial up or dedicated networks. Widely used for connections over modems.
Secure Sockets Layer	SSL	A protocol developed for transmitting private documents via the internet.
T1, Leased T1 Line		A dedicated phone connection supporting data rates of 1.544Mbits per second.
Transmission Control Protocol	TCP	A session oriented streaming protocol that provides for reliable communication between nodes on a network. Part of the TCP/IP Protocol Suite.
Transmission Control Protocol/Internet Protocol Suite	TCP/IP	Acronym used to designate a collection of multiple protocols originally developed by the US Department of Defense for computation between heterogeneous networks and computers. The standards are now guided by a collection of standards organizations, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB).
Wide Area Network	WAN	A computer network that spans a relatively large geographical area. Typically, a WAN consists of two or more local area networks (LANs).
Wide Area Network Router	WAN Router	