CBOE Application Programming Interface

Version 9.0.2

CBOE API Volume 4: CMi Dictionary of Attributes and Operations

Provides an alphabetic listing and definitions for all attributes and operations.

CBOE PROPRIETARY INFORMATION

15 July 2011

Document #[API-04]

Front Matter

Disclaimer

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 July 2011	V9.0.2	New constants for Wash Trade Prevention
		Added new structs and operations for cancel/replace of light orders and quote fill subscription changes (CMi V10 interfaces)
29 Apr 2011	V9.0.1	No changes
14 Jan 2011	V9.0	Added new structs, operations and error codes for Light Orders (CMi V9 interfaces)
23 Mar 2010	V7.0	New typedef for cmiUtil
05 Feb 2010	V7.0	Added CMi v8 interfaces
08 Jan 2010	V7.0	Added CMi v7 interfaces
07 Aug 2009	V6.1	New contingency types: BID_PEG_CROSS and OFFER_PEG_CROSS
22 May 2009	V6.0	New constants for Directed AIM
18 Jul 2008	V5.1	New constants for restricted series in order handling
29 Feb 2008	V5.0	New structs and operations for Cmi V5
18 Jan 2008	V4.2.4	New constants
02 Nov 2007	V4.2.3	New constants for cross-product spreads release (CBOEDIR_6.6)
01 June 2007	V4.2.2	New constants for CBSX billing enhancements
23 Feb 2007	V4.2.1	New values for:
		const cmiMarketData::MarketIndicator SlOW_ON_BID_SIDE = 33
		const cmiMarketData::MarketIndicator SlOW_ON_ASK_SIDE = 34
		const cmiMarketData::MarketIndicator SlOW_ON_BOTH_SIDE = 35
15 Dec 2006	V4.2	New cancel reason for User Input Monitor
08 Sept 2006	V4.1	New constants for Stock
25 May 2006	V4.0	New constants and definitions for Market

Date	Version	Description of Change
		Data Express (MDX)
06 Jan 2006	V3.2b	New constants and definitions for SAL and COF_MAIN
29 Jul 2005	V3.2	New EOP constants and modified definitions
08 Apr 2005	V3.1a	Documentation Errata
17 Dec 2004	V3.1	Updates for preferred DPM, internalizaiton and auctions
18 Jun 2004	V3.0	Updates for Hybrid and Stock
28 Apr 2004	V2.52	Constants and error code updates for the V2.5 production release.
10 Oct 2003	V2.62	New method descriptions for constants and intermarket messages.
29 Aug 2003	V2.61	Updated method descriptions.
31 Jul 2003	V2.6	Market linkage and definitions pertaining to Stock.
08 Jul 2003	V2.51	Updated/corrected definitions.
14 Mar 2003	V2.5	Support for Hybrid
24 Jan 2003	V2.1	Production Release Update to support Linkage P orders.
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 Release
14 Dec 2001	V2.0	Documentation Only Release
27 Apr 2001	V1.0b	Added Market Data role information
9 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.
11 Feb 2000	V0.7	Includes revisions to the API since the last update; updated/corrected definitions.
30 Sept 1999	V0.5	First release of this document.

About This Document

Purpose

This document provides an alphabetized listing of the data items in the CBOE Market Interface (CMi). It is designed to act as a reference to developers of applications programs that use the CMi.

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-01	CBOE API Volume 1: Overview and Concepts
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-05	CBOE API Volume 5: 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
FIX-01	CBOE API FIX Protocol Support Volume 1: Overview
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

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.

Typedefs

<typedef> cmiTraderActivity.ActivityFieldType	ActivityFieldType.short Number identifying the type of the field contained in the activity message - these types are not native types, but enumerated types that are defined within the CMi API.
<typedef> cmiUtil.ActivityReason	Identifies the source of an activity record. Activity records can be generated as a result of system or user actions.
<typedef> cmiTraderActivity.ActivityType	ActivityType.short Identifies the source of the activity record, whether the activity was initiated by the system or by the user (See cmiConstants::ActivityTypes).
<typedef> cmiIntermarketMessages.AlertResolution	
<typedef> cmiIntermarketMessages.AlertType	
<typedef> cmiOrder.AuctionState	The state of the auction (i.e. started). See cmiConstants::AuctionState
<typedef> cmiOrder.AuctionType	The auction type. See cmiConstants::AuctionType
<typedef> cmiMarketData.BookDepthUpdateType	Type of book depth update - indicates if updates is to insert, update, or delete a book depth entry.
<typedef> cmiOrder.CancelType	CancelType.short Type of cancellation action requested. Refers to how to treat the cancel quantity on a cancel request (See cmiConstants::CancelType).

<typedef> cmiProduct.ClassKey	ClassKey.Quote ClassKey.FloorTradeMaintenanceService ClassKey.MarketQuery ClassKey.Quote ClassKey.AuctionStruct ClassKey.MarketQuery ClassKey.long A unique number that is used to identify a product class.
<typedef> cmiSession.ClassState	ClassState.short State of class. State is currently maintained at the product (series) level. An interface to report state at the class level is provided for use with CBOE future CBOE markets.
<typedef> cmiProduct.CommodityQuantity	CommodityQuantity.double Quantity of the underlying commodity that the product is based on - for commodity derivatives only not option products.
<typedef> cmiOrder.ContingencyType	ContingencyType.short Order type contingency (Fill or kill, all or none, market on close, etc.) (See cmiConstants::ContingencyTypes)
<typedef> cmiOrder.Coverage	Coverage.char Specified type of coverage for an order. Can either be covered, uncovered, unspecified. See cmiConstants::CoverageTypes.
<typedef> cmiOrder.CrossingIndicator	CrossingIndicator.boolean Indicates if order is a part of a crossing order
<typedef > cmiUtil.TradingClassStatusIndicator	TradingClassStatusIndicator.short Indicates the status of a class pertaining to outages.
<typedef> cmiUtil.Description	

<typedef> cmiUtil.EntryType	EntryType.char Type of activity history.
<typedef> exceptions.ErrorCode	ErrorCode.long Code for an error being reported as an exception (See the cmiErrorCodes module).
<typedef> cmiUser.Exchange	 This field identifies the exchange for the current user.
<typedef> cmiIntermarketMessages.ExchangeMarketInfoType	ExchangeMarketInfoType.short
<typedef> cmiIntermarketMessages.ExecutionId	ExecutionId.string
<typedef> cmiMarketData.ExpectedOpeningPriceType	ExpectedOpeningPriceType.short Indicates if there will be an opening price for a product or if there is a market imbalance. If there is a market imbalance then the expected opening price will indicate if there are more buyers or more sellers. See cmiConstants::ExpectedOpeningPriceTypes.
<typedef> cmiProduct.ExpirationStyle	ExpirationStyle.char ExpirationStyle.char Style of Expiration (American or European) (See cmiConstants::ExpirationStyles).
<typedef> cmiIntermarketMessages.FillRejectReason	FillRejectReason.short
<typedef> cmiIntermarketMessages.HandlingInstruction	HandlingInstruction.short
<typedef> cmiMarketData.IntegerPrice	 Indicates the type of price. See cmiConstants::PriceConstants
<typedef> cmiMarketData.IntegerTime	

<typedef> cmiUtil.Key	
<typedef> cmiUtil.LinkageMechanism	Specifies a Linkage function. See (cmiConstants::ExtensionFields)
<typedef> cmiProduct.ListingState	ListingState.short Listing State of class (Active, Inactive, Obsolete, Unlisted) (refer to cmiConstants::ListingStates)
<typedef> cmiUser.LoginSessionMode	LoginSessionMode.char Mode of login session - Simulator, Network-Testing, Production
<typedef> cmiSession.LoginSessionType	LoginSessionType.UserAccessV6 LoginSessionType.UserAccessV3 LoginSessionType.UserAccessV4 LoginSessionType.short Indicates type of login being requested - either primary or secondary. Supports login to multiple CASs for fail over.
<typedef> cmiMarketData.MarketChangeReason	MarketChangeReason.short Reason market data was generated (See cmiConstants::MarketChangeReasons).
<typedef> cmiMarketData.MarketDataHistoryEntryType	MarketDataHistoryEntryType.short Indicates the type of market data history, such as quote, price report, expected opening price, market condition. See cmiConstants::MarketDataHistoryEntryTypes.
<typedef> cmiMarketData.MarketIndicator	Indicates the type of market indicator, such as regular, auto_execution, bid_is_book, etc.. See cmiConstants::MarketIndicators.
<typedef> cmiOrder.MatchType	Used to enter the price for internalization orders. The price can be auto matched, fixed limit price match or guaranteed starting price. Currently, limit price and auto match are the only two MatchTypes supported.

<typedef> cmiAdmin.MessageKey	MessageKey.long Unique Identifer of a message sent between client application and system application.
<typedef> cmiStrategy.MonthIncrement	 This field is the month increment for the strategy product.
<typedef> cmiMarketData.MultiplePartiesIndicator	
<typedef> cmiOrder.NBBOProtectionType	 This field determines if the order receives NBBO protection. The order has to be a customer order for this field to matter.
<typedef> cmiProduct.OptionType	OptionType.char Type of option - either 'C' - Call or 'P' - Put (See cmiConstants::OptionTypes).
<typedef> cmiIntermarketMessages.OrderBookTradableType	
<typedef> cmiUtil.OrderFlowDirection	 Specifies whether the order flow is inbound, outbound or both. (see cmiConstants::OrderFlowDirectionType)
<typedef> cmiOrder.OrderState	OrderState.short State of the order (Booked, canceled, etc.) (See cmiConstants::OrderStates
<typedef> cmiOrder.OriginType	OriginType.char Type of originator of the order - 'C'-Customer, 'F'-Firm, 'M'-Market Maker, 'B'-Broker (See cmiConstants::OrderOrigins).
<typedef> cmiMarketData.OverrideIndicatorType	
<typedef> cmiOrder.PositionEffect	PositionEffect.char Position Effect of the order (Opening or Closing) (See cmiConstants::PositionEffects)
<typedef> cmiIntermarketMessages.PreOpeningIndicationType	

<typedef> cmiProduct.PriceAdjustmentAction	PriceAdjustmentAction.short PriceAdjustmentAction.short Action to perform when applying a price adjustment to a product (see cmiConstants::PriceAdjustmentActions).
<typedef> cmiProduct.PriceAdjustmentType	PriceAdjustmentType.short PriceAdjustmentType.short Type of price adjustment (corporate action) (see cmiConstants::PriceAdjustmentTypes)
<typedef> cmiProduct.PriceDisplayType	PriceDisplayType.short PriceDisplayType.short Indicates if the price is to be displayed as a Fraction or a Decimal format.
<typedef> cmiUtil.PriceType	PriceType.short Type of price stored in PriceStruct (Market, Valued, No Price) (See cmiConstants::PriceTypes).
<typedef> cmiUtil.PricingModelParameter	PricingModelParameter.double Parameter for pricing model
<typedef> cmiProduct.ProductDescriptionName	ProductDescriptionName.string Name that identifies a particular ProductDescription. ProductDescriptions (combinations of attributes on prices and formats for products) are stored and referenced by name for ease in creation of new products.
<typedef> cmiProduct.ProductKey	ProductKey.MarketQuery ProductKey.AuctionStruct ProductKey.long Unique identifier for a product in the system. Applies to all products, including underlying products, not just products traded on the system.

<typedef> cmiSession.ProductState	ProductState.short State of a product within a trading session (such as, Closed, Open, etc.) (See cmiConstants::ProductStates).
<typedef> cmiProduct.ProductType	ProductType.short ProductType.AuctionStruct Type of product (debt, equity, option, etc.) (See cmiConstants::ProductTypes).
<typedef> cmiUtil.QueryDirection	QueryDirection.MarketQuery QueryDirection.MarketQuery QueryDirection.short Specifies the direction for retrieval of history information (see cmiConstants::QueryDirections).
<typedef> cmiUtil.QueueAction	QueueAction.CMIRecapConsumer QueueAction.CMITickerConsumer QueueAction.MarketQuery QueueAction.CMICurrentMarketConsumer QueueAction.MarketQuery QueueAction.CMICurrentMarketConsumer Specifies actions on a queue.
<typedef> cmiQuote.QuoteKey	QuoteKey.long Unique Identifier for a quote that is assigned by the system.
<typedef> cmiQuote.QuoteUpdateControl	
<typedef> cmiProduct.ReportingClassKey	ReportingClassKey.long Unique identifier of a reporting class.
<typedef> cmiUtil.ReportType	Type of report (fill, cancel, etc)

<typedef> cmiQuote.RFQType	RFQType.short Type of RFQ - Indicates if the RFQ was submitted manually or via an automated quotation system
<typedef> cmiUser.SessionNameType	 Identifies the session type used by the user.
<typedef> exceptions.SeverityLevel	SeverityLevel.short Severity Level (Not implemented yet).
<typedef> cmiUtil.Side	Side.char Either Bid or Ask (see cmiConstants::Sides).
<typedef> cmiUtil.Source	Source.char Specifies if the source is from the open outcry systems (TPF) or the electronic trading system (SBT) (See cmiConstants::Sources).
<typedef> cmiStrategy.StrategyType	StrategyType.short
<typedef> cmiProduct.Symbol	Symbol.string CBOE Ticker symbol for a product
<typedef> cmiOrder.TimeInForce	TimeInForce.char Time order is in force (Day, Good until cancelled, Good until date) (See cmiConstants::TimesInForce)
<typedef> cmiTrade.TradeSource	

<typedef> cmiSession.TradingSessionName	TradingSessionName.Quote TradingSessionName.FloorTradeMaintenanceService TradingSessionName.MarketQuery TradingSessionName.string TradingSessionName.AuctionStruct Name of the trading session. The trading session names are the same for both the CMi and the FIX interfaces. There is no enumeration for trading session names in the cmiConstants module because trading sessions are subject to change as trading at CBOE evolves. The getCurrentTradingSessions() operation is provided in cmi::Sessions to return the list of available trading session.
<typedef> cmiSession.TradingSessionState	TradingSessionState.short Trading Session State (Open or Closed) (See cmiConstants::TradingSessionStates).
<typedef> cmiUtil.UpdateStatusReason	UpdateStatusReason.short Reason for update. Refer to cmiConstants::StatusUpdateReasons for a list of possible values.
<typedef> cmiUser.UserRole	UserRole.char Each user is assigned to a role. The role defines what role the user plays in the market. The functions provided by the CAS can be restricted by role. At this time roles are defined for Market Makers, Brokers, Clearing Firms, and Help Desk. (See cmiConstants::UserRoles).
<typedef> cmiUtil.VersionLabel	VersionLabel.string Constant that specifies the CMI IDL Version.
<typedef> cmiMarketData.VolumeType	VolumeType.short Type of volume being reported (by contingency type) (See cmiConstants::VolumeTypes).

Struct Attribute

cmiUser.AccountStruct.account	Account to be associated with the user
cmiUser.AccountStruct.executingGiveupFirm	Executing or give up firm to be used for this user
cmiTraderActivity.ActivityFieldStruct.fieldName	Name of the field
cmiTraderActivity.ActivityFieldStruct.fieldType	Enumerated type of field (See cmiConstants::ActivityFieldType).
cmiTraderActivity.ActivityFieldStruct.fieldValue	Field value
cmiTraderActivity.ActivityHistoryStruct.activityRecords	Sequence of activity records
cmiTraderActivity.ActivityHistoryStruct.classKey	Unique identifier for a product class
cmiTraderActivity.ActivityHistoryStruct.endTime	Ending time of the sequence of activity records
cmiTraderActivity.ActivityHistoryStruct.startTime	Starting time of the sequence of activity records
cmiTraderActivity.ActivityRecordStruct.activityFields	Sequence of ActivityFieldStructs - contains the data associated with the activity. Each field struct is a name value pair that contains one data item for the activity record.
cmiTraderActivity.ActivityRecordStruct.entryType	Type of activity (See cmiConstants::ActivityTypes).
cmiTraderActivity.ActivityRecordStruct.eventTime	Time the activity occurred
cmiTraderActivity.ActivityRecordStruct.productKey	CBOE unique identifer for the product for which a trading activity occurred
cmiIntermarketMessages.AdminStruct.destinationExchange	Identifies the destination exchange.
cmiIntermarketMessages.AdminStruct.messageStruct	Information contained in the message. See cmiAdmin::MessageStruct
cmiIntermarketMessages.AdminStruct.productKey	Product identifier.
cmiIntermarketMessages.AdminStruct.sourceExchange	Identifies the exchange source.
cmiIntermarketMessages.AdminStruct.userAssignedId	User identifier.
cmiIntermarketMessages.AlertStruct.alertHdr	Alert header
cmiIntermarketMessages.AlertStruct.cboeMarketableOrder	Order is marketable against CBOE BBBO.
cmiIntermarketMessages.AlertStruct.comments	Comments pertaining to the alert.
cmiIntermarketMessages.AlertStruct.exchangeMarket	Information on the exchange. See cmiIntermarketMessages::ExchangeMarketStructSequence
cmiIntermarketMessages.AlertStruct.extensions	
cmiIntermarketMessages.AlertStruct.nbboAgentId	NBBO agent user ID.
cmiIntermarketMessages.AlertStruct.orderId	The order Id pertaining to the alert.
cmiIntermarketMessages.AlertStruct.productKeys	Product identifier.
cmiIntermarketMessages.AlertStruct.resolution	Resolution for the alert.

cmiIntermarketMessages.AlertStruct.tradeId	Identifies the trade. See cmiUtil::CboeIdStruct.
cmiIntermarketMessages.AlertStruct.updatedById	Tracks the last user who updated the field.
cmiOrder.AuctionStruct.auctionedOrderContingencyType	The contingency type of the auctioned order.
cmiOrder.AuctionStruct.auctionID	Unique identifier of the auction.
cmiOrder.AuctionStruct.auctionQuantity	The order quantity in the auction.
cmiOrder.AuctionStruct.auctionState	The state (i.e. Started) of the auction.
cmiOrder.AuctionStruct.auctionType	Type of auction supported by CBOEdirect.
cmiOrder.AuctionStruct.classKey	A unique number that is used to identify a product class.
cmiOrder.AuctionStruct.entryTime	The time the order entered the auction.
cmiOrder.AuctionStruct.extensions	Used to persist arbitrary data along with the auction.
cmiOrder.AuctionStruct.productKey	Unique identifier for a product in the system. Applies to all products, including underlying products, not just products traded on the system
cmiOrder.AuctionStruct.productType	Type of product (debt, equity, option, etc.) (See cmiConstants::ProductTypes).
cmiOrder.AuctionStruct.sessionName	Name of the trading session where the user permits the order to be traded.
cmiOrder.AuctionStruct.side	Either Bid or Ask (see cmiConstants::Sides).
cmiOrder.AuctionStruct.startingPrice	The auction start price.
cmiOrder.AuctionSubscribeResultStruct.auctionType	The type of auction subscription.
cmiOrder.AuctionSubscribeResultStruct.subscriptionResult	The result of the subscription. See cmiUtil.
cmiIntermarketMessages.BookDepthDetailedStruct.buyOrdersAtDifferentPrice	Buy order prices.
cmiIntermarketMessages.BookDepthDetailedStruct.productKeys	Products identifier.
cmiIntermarketMessages.BookDepthDetailedStruct.sellOrdersAtDifferentPrice	Sell order prices.
cmiIntermarketMessages.BookDepthDetailedStruct.sessionName	Trading session name.
cmiIntermarketMessages.BookDepthDetailedStruct.transactionSequenceNumber	Transaction identifier.
cmiMarketData.BookDepthStruct.allPricesIncluded	Indicates if all prices from the order book have been set to true. If false the book depth is truncated due to message size restrictions.
cmiMarketData.BookDepthStruct.buySideSequence	Sequence of prices on the buy side of the market.
cmiMarketData.BookDepthStruct.productKeys	Product Key of the product whose order book update is being reported.
cmiMarketData.BookDepthStruct.sellSideSequence	A sequence of prices on the ask side of the market.
cmiMarketData.BookDepthStruct.sessionName	Name of the trading session from which this market data originated.
cmiMarketData.BookDepthStruct.transactionSequenceNumber	Sequence number of the book depth update. Used by the client application for message sequencing.

cmiMarketData.BookDepthStructV2.allPricesIncluded	Indicates if all prices from the order book have been set to true. If false the book depth is truncated due to message size restrictions.
cmiMarketData.BookDepthStructV2.buySideSequence	Sequence of prices on the buy side of the market.
cmiMarketData.BookDepthStructV2.productKeys	Product Key of the product whose order book update is being reported.
cmiMarketData.BookDepthStructV2.sellSideSequence	A sequence of prices on the ask side of the market.
cmiMarketData.BookDepthStructV2.sessionName	Name of the trading session from which this market data originated.
cmiMarketData.BookDepthStructV2.transactionSequenceNumber	Sequence number of the book depth update. Used by the client application for message sequencing.
cmiMarketData.BookDepthUpdatePriceStruct.contingencyVolume	Portion of the total volume that represents contingency orders.
cmiMarketData.BookDepthUpdatePriceStruct.price	Price to be updated in the book.
cmiMarketData.BookDepthUpdatePriceStruct.totalVolume	Total volume at this price.
cmiMarketData.BookDepthUpdatePriceStruct.updateType	Type of update.
cmiMarketData.BookDepthUpdateStruct.buySideChanges	Changes in quantity and/or price in the book on the Buy side.
cmiMarketData.BookDepthUpdateStruct.exchange	Exchange identifier where the product trades.
cmiMarketData.BookDepthUpdateStruct.productKeys	Identifier of the product for which this book depth is issued.
cmiMarketData.BookDepthUpdateStruct.sellSideChanges	Changes in quantity and/or price in the book on the Sell side.
cmiMarketData.BookDepthUpdateStruct.sequenceNumber	Unique serial number that identifies this book depth update.
cmiMarketData.BookDepthUpdateStruct.sessionName	Name of the trading session for which the book depth resides.
cmiOrder.BustReinstateReportStruct.bustedQuantity	The busted trade quantity
cmiOrder.BustReinstateReportStruct.price	Trade price at which the bust trade was restated
cmiOrder.BustReinstateReportStruct.productKey	Unique identifier of the product for which a busted trade was reinstated.
cmiOrder.BustReinstateReportStruct.reinstatedQuantity	Quantity that was reinstated
cmiOrder.BustReinstateReportStruct.sessionName	Name of the trading session in which the busted trade was reinstated.
cmiOrder.BustReinstateReportStruct.side	Side of the trade for this user
cmiOrder.BustReinstateReportStruct.timeSent	Timestamp when the bust reinstate report was issued
cmiOrder.BustReinstateReportStruct.totalRemainingQuantity	The total remaining quantity after the bust trade was reinstated
cmiOrder.BustReinstateReportStruct.tradeId	Trade Id that was reinstated
cmiOrder.BustReinstateReportStruct.transactionSequenceNumber	Transaction sequence number for the bust trade reinstatement report
cmiOrder.BustReportStruct.bustedQuantity	Quantity of the trade that was busted
cmiOrder.BustReportStruct.bustReportType	Type of trade bust report.
cmiOrder.BustReportStruct.executingOrGiveUpFirm	Executing or Give Up Firm associated with the trade

cmiOrder.BustReportStruct.price	Price of the busted trade
cmiOrder.BustReportStruct.productKey	Product Key of the product of the trade that was busted
cmiOrder.BustReportStruct.reinstateRequestedQuantity	Quantity of the busted trade that the user wants to reinstate
cmiOrder.BustReportStruct.sessionName	Name of the trading session in which the trade was busted.
cmiOrder.BustReportStruct.side	Side of the trade that was busted
cmiOrder.BustReportStruct.timeSent	Time the bust report was sent
cmiOrder.BustReportStruct.tradeId	Trade identifier
cmiOrder.BustReportStruct.transactionSequenceNumber	Unique transaction number for this report
cmiOrder.BustReportStruct.userAcronym	The user acronym for the trade bust.
cmiOrder.BustReportStruct.userId	User associated with the trade
cmiUtil.CallbackInformationStruct.ior	The stringified Interoperable Object Reference for the callback object described in the CallbackInformationStruct.
cmiUtil.CallbackInformationStruct.subscriptionInterface	The name of the interface of the callback object that is described in the CallbackInformationStruct.
cmiUtil.CallbackInformationStruct.subscriptionOperation	Name of the operation on the callback object that is being described in the CallbackInformationStruct. This is usually the operation that did not provide a response to the CAS - prompting the callback object to be deregistered.
cmiUtil.CallbackInformationStruct.subscriptionValue	This is a string representation of the data for which the callback object was subcribed.
cmiOrder.CancelReportStruct.cancelledQuantity	Cancelled quantity
cmiOrder.CancelReportStruct.cancelReason	Sets a reason for a cancel report.
cmiOrder.CancelReportStruct.cancelReportType	Identifies the type of cancel report.
cmiOrder.CancelReportStruct.mismatchedQuantity	Quantity difference between what the client thought the remaining quantity was at the time of the cancel request and the actual remaining quantity when the cancel request was received by the trading system
cmiOrder.CancelReportStruct.orderId	Identifier for the cancelled order
cmiOrder.CancelReportStruct.orsId	ORS System ID for the order.
cmiOrder.CancelReportStruct.sessionName	Name of the trading session in which an order was fully or partially canceled.
cmiOrder.CancelReportStruct.timeSent	Time cancel report was sent by system to user.
cmiOrder.CancelReportStruct.tlcQuantity	Quantity that was too late to cancel - most likely due to a fill that took place prior to the cancel request being received by the system
cmiOrder.CancelReportStruct.totalCancelledQuantity	Total quantity cancelled on order

cmiOrder.CancelReportStruct.transactionSequenceNumber	Transaction number assigned to the message to guarantee uniqueness and ordering between CAS and Exchange Services. Can be ignored by client application, as the CAS guarantees ordering.
cmiOrder.CancelReportStruct.userAssignedCancelId	The user assigned id that was entered by the user on the cancel request.
cmiOrder.CancelRequestStruct.cancelReason	Reason for the cancel request
cmiOrder.CancelRequestStruct.cancelType	The type of cancellation requested. Specify to cancel all, cancel a specific quantity or specify the remaining quantity desired after the cancel.
cmiOrder.CancelRequestStruct.orderId	Identifier of order to be cancelled.
cmiOrder.CancelRequestStruct.quantity	Quanity to be cancelled or remaining quantity based upon the cancel type (See cmiConstants::OrderCanceTypes).
cmiOrder.CancelRequestStruct.sessionName	Name of the trading session in which the order is to be canceled.
cmiOrder.CancelRequestStruct.userAssignedCancelId	User assigned identifier for this cancel request. Recommend that this id follow branch + branch sequence number semantics.
cmiUtil.CboeIdStruct.highCboeId	Major part of the key (most significant bits).
cmiUtil.CboeIdStruct.lowCboeId	Minor part of key (least significant bits).
cmiQuote.ClassQuoteResultStruct.errorCode	Error code if error was encountered for a particular series in the class. Refer to cmiErrorCodes::DataValidationCodes and cmiErrorCodes::TransactionFailedCodes for a list of enumerations that can apply to quote rejection.
cmiQuote.ClassQuoteResultStruct.productKey	Product key of the product for which the quote error was encountered
cmiQuote.ClassQuoteResultStructV2.errorCode	
cmiQuote.ClassQuoteResultStructV2.productKey	Product key of the product for which the quote error was encountered.
cmiQuote.ClassQuoteResultStructV2.quoteKey	Quote key of the product for which the quote error was encountered.
cmiQuote.ClassQuoteResultStructV3.quoteResult	Result of the quote.
cmiQuote.ClassQuoteResultStructV3.quoteUpdateControlId	Identification for quote update.
cmiSession.ClassStateStruct.classKey	Unique identifier of the class
cmiSession.ClassStateStruct.classState	State of the class. NOT CURRENTLY IN USE.
cmiSession.ClassStateStruct.classStateTransactionSequenceNumber	Sequence number used to order class status information.
cmiSession.ClassStateStruct.sessionName	Name of the trading session where the class is active.
cmiProduct.ClassStruct.activationDate	Activation date.
cmiProduct.ClassStruct.classKey	Class Key (Unique Identifier).
cmiProduct.ClassStruct.classSymbol	Class Symbol.
cmiProduct.ClassStruct.createdTime	Time class was created in the system.
cmiProduct.ClassStruct.epwFastMarketMultiplier	Fast market multiplier that is to be used for quote bid-ask spreads under fast market conditions

cmiProduct.ClassStruct.epwValues	A table of bid-ask spread widths based upon price ranges in the bid prices
cmiProduct.ClassStruct.inactivationDate	Inactivation date.
cmiProduct.ClassStruct.lastModifiedTime	Last time class information was modified.
cmiProduct.ClassStruct.listingState	Listing state of the product, such as active, inactive, obsolete, unlisted (see cmiConstants::ListingStates)
cmiProduct.ClassStruct.primaryExchange	Primary exchange class is traded on.
cmiProduct.ClassStruct.productDescription	Product details
cmiProduct.ClassStruct.productType	Type of product (equity, future, option, etc.) (Refer to cmiConstants::ProductTypes).
cmiProduct.ClassStruct.reportingClasses	List of reporting classes.
cmiProduct.ClassStruct.testClass	Test classes in the system that are not real products.
cmiProduct.ClassStruct.underlyingProduct	Underlying Product of products belonging to this class.
cmiOrder.ContraPartyStruct.firm	Member Firm of contraparty (Executing or Giveup Firm)
cmiOrder.ContraPartyStruct.quantity	Quantity traded with this contraparty
cmiOrder.ContraPartyStruct.user	User acronym of contra party.
cmiOrder.CrossOrderStruct.buySideOrder	Contains the buy side information of a cross order.
cmiOrder.CrossOrderStruct.sellSideOrder	Contains the sell side information of the cross order.
cmiIntermarketMessages.CurrentIntermarketBestStruct.askPrice	Current best intermarket ask price.
cmiIntermarketMessages.CurrentIntermarketBestStruct.askVolume	The intermarket ask volume.
cmiIntermarketMessages.CurrentIntermarketBestStruct.bidprice	Current intermarket bid price.
cmiIntermarketMessages.CurrentIntermarketBestStruct.bidVolume	The intermarket bid volume.
cmiIntermarketMessages.CurrentIntermarketBestStruct.exchange	Exchange identifier.
cmiIntermarketMessages.CurrentIntermarketBestStruct.marketCondition	The type of market condition reported.
cmiIntermarketMessages.CurrentIntermarketBestStruct.sentTime	Time the intermarket information was sent from the trading system.
cmiIntermarketMessages.CurrentIntermarketStruct.otherMarketsBest	A sequence of CurrentIntermarketBestStructs.
cmiIntermarketMessages.CurrentIntermarketStruct.productKeys	Identifier of the intermarket product.
cmiMarketData.CurrentMarketStruct.askIsMarketBest	Indicates if the ask price is the market best.
cmiMarketData.CurrentMarketStruct.askPrice	Current best Ask Price
cmiMarketData.CurrentMarketStruct.askSizeSequence	Sequence of MarketVolumeStructs that contain the order volume broken down by order contingency type at the current market.
cmiMarketData.CurrentMarketStruct.bidIsMarketBest	Indicates if current market price is the market best.
cmiMarketData.CurrentMarketStruct.bidPrice	Current best Bid Price
cmiMarketData.CurrentMarketStruct.bidSizeSequence	Sequence of MarketVolumeStructs that contain the order volume broken down by order contingency type at the current market.
cmiMarketData.CurrentMarketStruct.exchange	Exchange from which the bid originated.

cmiMarketData.CurrentMarketStruct.legalMarket	Indicates if the current market bid-ask spread is a legal market per CBOE regulation
cmiMarketData.CurrentMarketStruct.productKeys	Product Keys for the product whose current market is being reported
cmiMarketData.CurrentMarketStruct.sentTime	Time market information was sent from trading system.
cmiMarketData.CurrentMarketStruct.sessionName	Name of trading session from which the current market message originated
cmiMarketData.CurrentMarketStructV2.askIsMarketBest	Ask price is the market best. Set to "true" indicates that the ask price is part of the NBBO. Set to "false" indicates that the ask price may not be part of the NBBO.
cmiMarketData.CurrentMarketStructV2.bidIsMarketBest	The bid price is the market best. Set to "true" indicates that the bid price is part of the NBBO. Set to "false" indicates that the bid price may not be part of the NBBO.
cmiMarketData.CurrentMarketStructV2.currentMarketViews	Sequence for current market views.
cmiMarketData.CurrentMarketStructV2.exchange	Name of the exchange.
cmiMarketData.CurrentMarketStructV2.legalMarket	In a legal market.
cmiMarketData.CurrentMarketStructV2.productKeys	Product Keys for the product whose current market is being reported.
cmiMarketData.CurrentMarketStructV2.sentTime	Time the information was sent.
cmiMarketData.CurrentMarketStructV2.sessionName	Session name for the product that is being reported.
cmiMarketData.CurrentMarketStructV4.askPrice	The ask price.
cmiMarketData.CurrentMarketStructV4.askSizeSequence	Sequence of MarketVolumeStructs that contain the order volume broken down by order contingency type at the current market.
cmiMarketData.CurrentMarketStructV4.bidPrice	The bid price
cmiMarketData.CurrentMarketStructV4.bidSizeSequence	Sequence of MarketVolumeStructs that contain the order volume broken down by order contingency type at the current market.

cmiMarketData.CurrentMarketStructV4.bidTickDirection	The direction of change for the bid PLUS_TICK '+' Means the current last sale price is higher than the previous last sale price. Also known as an "uptick". MINUS_TICK '-' Means the current last sale price is lower than the previous last sale price. Also known as a "downtick". ZERO_MINUS_TICK '_' Means the current last sale price is the same as the previous last sale price, but is lower than the most recent different last sale price. ZERO_PLUS_TICK '*' Means the current last sale price is the same as the previous last sale price, but is higher than the most recent different last sale price. UNKNOWN_TICK ' 'Means the tick direction field is not used or not known.
cmiMarketData.CurrentMarketStructV4.classKey	Class key for the product whose current market is being reported.
cmiMarketData.CurrentMarketStructV4.currentMarketType	The market type, either best market or best public market.
cmiMarketData.CurrentMarketStructV4.exchange	Name of the exchange.
cmiMarketData.CurrentMarketStructV4.marketIndicator	The market indicator for this product.
cmiMarketData.CurrentMarketStructV4.priceScale	The number of decimal places to use in conjunction with the integer prices.
cmiMarketData.CurrentMarketStructV4.productKey	Product key for the product whose current market is being reported.
cmiMarketData.CurrentMarketStructV4.productState	The product state for this product.
cmiMarketData.CurrentMarketStructV4.productType	The product type (equity, option, etc.)
cmiMarketData.CurrentMarketStructV4.sentTime	Time that the information was sent.
cmiMarketData.CurrentMarketViewStruct.askPrice	The ask price.
cmiMarketData.CurrentMarketViewStruct.askSizeSequence	A sequence of the ask size.
cmiMarketData.CurrentMarketViewStruct.bidPrice	The bid price.
cmiMarketData.CurrentMarketViewStruct.bidSizeSequence	A sequence of the bid size.
cmiMarketData.CurrentMarketViewStruct.currentMarketViewType	The type of current market view being reported.
cmiUtil.DateStruct.day	value between 1 and 31

cmiUtil.DateStruct.month	value between 1 and 12
cmiUtil.DateStruct.year	value of year in YYYY format
cmiUtil.DateTimeStruct.date	refer to DateStruct
cmiUtil.DateTimeStruct.time	refer to TimeStruct
cmiUser.DpmStruct.dpmAssignedClasses	List of classes that have been assigned to the Designated Primary Market Maker.
cmiUser.DpmStruct.dpmUserId	User id of the Designated Primary Market Maker.
cmiProduct.EPWStruct.maximumAllowableSpread	Bid-ask spread that is in effect when the bid price is within the [MinimumBidRange, MaximumBidRange]
cmiProduct.EPWStruct.maximumBidRange	Maximum bid for which this bid-ask spread applies
cmiProduct.EPWStruct.minimumBidRange	Minimum bid for which this bid-ask spread applies
exceptions.ExceptionDetails.dateTime	Date time of exception as a string
exceptions.ExceptionDetails.error	Error number (See cmiErrorCodes module).
exceptions.ExceptionDetails.message	Description of error
exceptions.ExceptionDetails.severity	Severity level - not implemented at this time.
cmiUser.ExchangeAcronymStruct.acronym	Exchange acronym identifier.
cmiUser.ExchangeAcronymStruct.exchange	Exchange identifier.
cmiUser.ExchangeFirmStruct.exchange	Exchange identifier.
cmiUser.ExchangeFirmStruct.firmNumber	Firm identifier.
cmiMarketData.ExchangeIndicatorStruct.exchange	The name of the exchange.
cmiMarketData.ExchangeIndicatorStruct.marketCondition	The condition of the market.
cmiIntermarketMessages.ExchangeMarketStruct.askExchangevolumes	A series of exchange names and volumes.
cmiIntermarketMessages.ExchangeMarketStruct.bestAskPrice	Exchange(s) best ask price.
cmiIntermarketMessages.ExchangeMarketStruct.bestBidPrice	Exchange(s) best bid price.
cmiIntermarketMessages.ExchangeMarketStruct.bidExchangeVolumes	A series of exchange names and volume.
cmiIntermarketMessages.ExchangeMarketStruct.marketInfoType	Exchange market information.
cmiMarketData.ExchangeVolumeStruct.exchange	Exchange identifier
cmiMarketData.ExchangeVolumeStruct.volume	Volume in contracts for options or futures, shares for stock
cmiMarketData.ExpectedOpeningPriceStruct.eopType	Expected opening price type. Value will be set to Opening Price, More Buyers, or More Sellers. See the cmiConstants::ExpectedOpeningPriceTypes for more information.
cmiMarketData.ExpectedOpeningPriceStruct.expectedOpeningPrice	Expected Opening Price
cmiMarketData.ExpectedOpeningPriceStruct.imbalanceQuantity	Quantity of any order imbalance

cmiMarketData.ExpectedOpeningPriceStruct.legalMarket	Used to indicate if expected opening prices are within exchange prescribed bid-ask spread
cmiMarketData.ExpectedOpeningPriceStruct.productKeys	Product whose expected opening price is being reported
cmiMarketData.ExpectedOpeningPriceStruct.sessionName	Name of the trading session for which the expected opening price of a product was calculated
cmiTrade.ExternalAtomicTradeEntryStruct.buyerAccount	The buyer side account number.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerBroker	Buyer side broker acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerCmta	The CMTA for the buyer side of the trade.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerFirm	Buyer side Firm acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerOptionalData	Buyer side optional data. This is an optional field.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerOriginator	Buyer side originator acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerOriginType	The buyer side origin.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerPositionEffect	The buyer side position.
cmiTrade.ExternalAtomicTradeEntryStruct.buyerSubaccount	Buyer side subaccount
cmiTrade.ExternalAtomicTradeEntryStruct.entryTime	The time the external trade was entered.
cmiTrade.ExternalAtomicTradeEntryStruct.entryType	Type of trade entered in the system.
cmiTrade.ExternalAtomicTradeEntryStruct.quantity	Trade quantity.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerAccount	The seller's account number.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerBroker	The seller side broker acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerCmta	The seller side CMTA.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerFirm	Seller side Firm acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerOptionalData	Seller side optional data. This field is optional.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerOriginator	Seller side originator acronym.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerOriginType	The seller side origin.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerPositionEffect	The seller side position.
cmiTrade.ExternalAtomicTradeEntryStruct.sellerSubaccount	The seller side subaccount.
cmiTrade.ExternalAtomicTradeEntryStruct.sessionName	The name of the trading session where the trade took place.
cmiTrade.ExternalAtomicTradeResultStruct.active	Shows if trade is active.
cmiTrade.ExternalAtomicTradeResultStruct.atomicTradeId	Atomic trade identification.
cmiTrade.ExternalAtomicTradeResultStruct.buyerAccount	The buyer side account number.
cmiTrade.ExternalAtomicTradeResultStruct.buyerBroker	The buyer side broker acronym.
cmiTrade.ExternalAtomicTradeResultStruct.buyerCmta	The CMTA for the buyer side of the trade.
cmiTrade.ExternalAtomicTradeResultStruct.buyerCorrespondentId	The buyer side correspondent ID.
cmiTrade.ExternalAtomicTradeResultStruct.buyerFirm	The buyer side Firm acronym.

cmiTrade.ExternalAtomicTradeResultStruct.buyerFirmBranch	The buyer side Firm branch number. Used for orders only.
cmiTrade.ExternalAtomicTradeResultStruct.buyerFirmBranchSequence Number	The buyer side branch sequence number. Used for orders only.
cmiTrade.ExternalAtomicTradeResultStruct.buyerOptionalData	The buyer side optional data field.
cmiTrade.ExternalAtomicTradeResultStruct.buyerOrderOrQuote	The buyer side order or quote.
cmiTrade.ExternalAtomicTradeResultStruct.buyerOrderOrQuoteKey	The buyer side order or quote identifier.
cmiTrade.ExternalAtomicTradeResultStruct.buyerOriginator	The buyer side originator acronym.
cmiTrade.ExternalAtomicTradeResultStruct.buyerOriginType	The buyer side origin.
cmiTrade.ExternalAtomicTradeResultStruct.buyerPositionEffect	The buyer side position.
cmiTrade.ExternalAtomicTradeResultStruct.buyerSubaccount	The buyer side subaccount.
cmiTrade.ExternalAtomicTradeResultStruct.entryTime	Trade entry time.
cmiTrade.ExternalAtomicTradeResultStruct.entryType	Type of trade entered in the system.
cmiTrade.ExternalAtomicTradeResultStruct.lastEntryType	Collection of last entry types.
cmiTrade.ExternalAtomicTradeResultStruct.lastUpdateTime	Last time the trade was updated.
cmiTrade.ExternalAtomicTradeResultStruct.matchedSequenceNumber	The sequence number for the match trade.
cmiTrade.ExternalAtomicTradeResultStruct.quantity	The trade quantity.
cmiTrade.ExternalAtomicTradeResultStruct.reinstatableForBuyer	Sets buyer side reinstateable trade.
cmiTrade.ExternalAtomicTradeResultStruct.reinstatableForSeller	Sets seller side reinstatable trade.
cmiTrade.ExternalAtomicTradeResultStruct.sellerAccount	The seller side account.
cmiTrade.ExternalAtomicTradeResultStruct.sellerBroker	The seller side broker acronym.
cmiTrade.ExternalAtomicTradeResultStruct.sellerCmta	Seller side CMTA.
cmiTrade.ExternalAtomicTradeResultStruct.sellerCorrespondentId	Seller side correspondent identification.
cmiTrade.ExternalAtomicTradeResultStruct.sellerFirm	The seller side Firm acronym.
cmiTrade.ExternalAtomicTradeResultStruct.sellerFirmBranch	The seller side Firm branch number. Used for orders only.
cmiTrade.ExternalAtomicTradeResultStruct.sellerFirmBranchSequenceN umber	Seller side Firm branch sequence number. Used for orders only.
cmiTrade.ExternalAtomicTradeResultStruct.sellerOptionalData	The seller side optional data.
cmiTrade.ExternalAtomicTradeResultStruct.sellerOrderOrQuote	Seller side order or quote.
cmiTrade.ExternalAtomicTradeResultStruct.sellerOrderOrQuoteKey	Seller side order or quote identifier.
cmiTrade.ExternalAtomicTradeResultStruct.sellerOriginator	The seller side originator acronym.
cmiTrade.ExternalAtomicTradeResultStruct.sellerOriginType	Seller side origin.
cmiTrade.ExternalAtomicTradeResultStruct.sellerPositionEffect	Seller side position.
cmiTrade.ExternalAtomicTradeResultStruct.sellerSubaccount	The seller side subaccount.
cmiTrade.ExternalAtomicTradeResultStruct.sessionName	The name of the trading session where the trade took place.

cmiTrade.ExternalBustTradeStruct.atomicTradeId	Identifier of the atomic trade.
cmiTrade.ExternalBustTradeStruct.bustedQuantity	The bust trade quantity.
cmiTrade.ExternalBustTradeStruct.buyerReinstateRequested	Sets buyer trade reinstate.
cmiTrade.ExternalBustTradeStruct.sellerReinstateRequested	Sets buyer trade reinstate.
cmiTrade.ExternalTradeEntryStruct.asOffFlag	
cmiTrade.ExternalTradeEntryStruct.businessDate	The business day of the external trade.
cmiTrade.ExternalTradeEntryStruct.bustable	Defines external trade as bustable.
cmiTrade.ExternalTradeEntryStruct.externalTradeType	Type of external trade.
cmiTrade.ExternalTradeEntryStruct.handlingInstruction	Specific trade instructions.
cmiTrade.ExternalTradeEntryStruct.name	
cmiTrade.ExternalTradeEntryStruct.parties	Trade participants.
cmiTrade.ExternalTradeEntryStruct.price	The price of the external trade.
cmiTrade.ExternalTradeEntryStruct.productKey	Product identifier.
cmiTrade.ExternalTradeEntryStruct.quantity	The quantity amount of the trade.
cmiTrade.ExternalTradeEntryStruct.sessionName	The trading session for the external trade entry.
cmiTrade.ExternalTradeEntryStruct.settlementDate	The trade settlement date.
cmiTrade.ExternalTradeEntryStruct.theTradeSource	External trade source.
cmiTrade.ExternalTradeEntryStruct.timeTraded	The time of the trade.
cmiTrade.ExternalTradeEntryStruct.transactionTime	The time of the trade transaction.
cmiTrade.ExternalTradeReportStruct.asOfFlag	
cmiTrade.ExternalTradeReportStruct.businessDate	The date the trade occurred.
cmiTrade.ExternalTradeReportStruct.bustable	Sets the trade to be bustable or unbustable.
cmiTrade.ExternalTradeReportStruct.externalTradeType	The type of external trade.
cmiTrade.ExternalTradeReportStruct.parties	Trade participants.
cmiTrade.ExternalTradeReportStruct.price	The external trade price.
cmiTrade.ExternalTradeReportStruct.productKey	Product identifier.
cmiTrade.ExternalTradeReportStruct.quantity	The trade quantity for the external trade report.
cmiTrade.ExternalTradeReportStruct.sessionName	The trading session where the trade occurred.
cmiTrade.ExternalTradeReportStruct.settlementDate	The trade settlement date.
cmiTrade.ExternalTradeReportStruct.theTradeSource	External trade source.
cmiTrade.ExternalTradeReportStruct.timeTraded	The time the trade occurred.

cmiTrade.ExternalTradeReportStruct.tradeId	Trade identification.
cmiTrade.ExternalTradeReportStruct.transactionTime	The time the trade transaction occurred.
cmiOrder.FilledReportStruct.account	The account field has multiple uses based upon type of market participant. For customers and firms it is used for customer account information. For market makers it contains the clearing subaccount information.
cmiOrder.FilledReportStruct.cmta	Clearing Member Trade Agreement - used if a different clearing member firm is assigned to the order.
cmiOrder.FilledReportStruct.contraParties	Contra parties that were traded against (includes the volume for each contra party)
cmiOrder.FilledReportStruct.executingBroker	Executing broker acrynom for trade.
cmiOrder.FilledReportStruct.executingOrGiveUpFirm	Member Firm Acronym who is the executing or give up firm.
cmiOrder.FilledReportStruct.extensions	Internal CBOE field for extensions to the order message.
cmiOrder.FilledReportStruct.fillReportType	Type of fill report requested.
cmiOrder.FilledReportStruct.leavesQuantity	Quantity remaining in order that has not been traded

cmiOrder.FilledReportStruct.optionalData	Optional data entered by the Originator when the order was entered. Optional text field. Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional. For Sweep and AIM: Enter A:AIS in the primary order to instruct CBOE to sweep all better priced protected quotes at away exchanges at the same time as the commencement of the AIM auction. For AIM Auctions: Enter A:AIM as the first characters. If a Firm does not wish to cancel the primary order when the auction expires, the Firm must enter A:AIR, instead of A:AIM, in this field. This will designate the primary order to be returned to the system and trade or book as a regular order. Options: For Customer orders, the first four characters of this field get reported to the last four characters of the CBOE Trade Match Optional Data field. These four characters get reported to the OCC. Preferred DPM: Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional
cmiOrder.FilledReportStruct.originator	The Member Firm or Organization that originated the Order.
cmiOrder.FilledReportStruct.orsId	Order Routing System Identifier
cmiOrder.FilledReportStruct.positionEffect	Position effect (opening or closing or none)
cmiOrder.FilledReportStruct.price	Price at which this portion of the order was traded
cmiOrder.FilledReportStruct.productKey	Product key of the product that was traded.
cmiOrder.FilledReportStruct.sessionName	Name of the trading session in which the trade occurred.
cmiOrder.FilledReportStruct.side	Side of the order (Bid or Ask)
cmiOrder.FilledReportStruct.subaccount	For market makers used to populate the clearing optional data.
cmiOrder.FilledReportStruct.timeSent	Time the trade was executed by the trading system
cmiOrder.FilledReportStruct.tradedQuantity	Quantity traded

cmiOrder.FilledReportStruct.tradeId	Identifier of the trade. Systems should track trades using this identifier, as it is referenced on trade busts and trade bust reinstates. Firms should also keep track of the OrderId and transactionSequenceNumber to ensure that the fill report message is unique.
cmiOrder.FilledReportStruct.transactionSequenceNumber	Transaction Sequence Number assigned to the report to help the client application order the fill report.
cmiOrder.FilledReportStruct.userAcronym	The acronym for a user.
cmiOrder.FilledReportStruct.userAssignedId	User assigned identifier for quotes. Provided for FIX compatibility.
cmiOrder.FilledReportStruct.userId	User id of the user that submitted the order.
cmiIntermarketMessages.FillRejectRequestStruct.fillReportExtensions	Extensions to the order message.
cmiIntermarketMessages.FillRejectRequestStruct.orderId	Identifier of order to reject fill.
cmiIntermarketMessages.FillRejectRequestStruct.rejectReason	Reject reason.
cmiIntermarketMessages.FillRejectRequestStruct.tradedQuantity	Quantity traded
cmiIntermarketMessages.FillRejectRequestStruct.tradePrice	
cmiIntermarketMessages.FillRejectStruct.extensions	Extensions to the order message.
cmiIntermarketMessages.FillRejectStruct.order	Order information.
cmiIntermarketMessages.FillRejectStruct.rejectReason	Reject reason.
cmiIntermarketMessages.FillRejectStruct.text	Fill reject text.
cmiIntermarketMessages.FillRejectStruct.tradeId	Unique identifier that identifies the trade.
cmiIntermarketMessages.FillRejectStruct.transactionSequenceNumber	Transaction sequence number for the fill reject.
cmiTrade.FloorTradeEntryStruct.account	Firm account number.
cmiTrade.FloorTradeEntryStruct.cmta	The Firm's CMTA.
cmiTrade.FloorTradeEntryStruct.contraBroker	The acronym of the Contra Broker.
cmiTrade.FloorTradeEntryStruct.contraFirm	The Firm acronym of the Contra Broker.
cmiTrade.FloorTradeEntryStruct.executingMarketMaker	The acronym of the Market Maker executing the trade.
cmiTrade.FloorTradeEntryStruct.firm	Firm acronym.
cmiTrade.FloorTradeEntryStruct.optionalData	Optional data field
cmiTrade.FloorTradeEntryStruct.positionEffect	
cmiTrade.FloorTradeEntryStruct.price	The trade price.
cmiTrade.FloorTradeEntryStruct.productKey	Product identifier.
cmiTrade.FloorTradeEntryStruct.quantity	Trade quantity.
cmiTrade.FloorTradeEntryStruct.sessionName	The trading session where the floor trade occurred.
cmiTrade.FloorTradeEntryStruct.side	The buy or sell side of the trade .
cmiTrade.FloorTradeEntryStruct.subaccount	Firm subaccount.

cmiTrade.FloorTradeEntryStruct.timeTraded	The time the floor trade occurred.
cmiAdmin.HeartBeatStruct.currentDate	Current Date reported by CAS
cmiAdmin.HeartBeatStruct.currentTime	Current Time reported by the CAS.
cmiAdmin.HeartBeatStruct.pulseInterval	Interval between heartbeats in seconds
cmiAdmin.HeartBeatStruct.requestID	Contains the request ID that generated the heartbeat
cmiIntermarketMessages.HeldOrderCancelReportStruct.cancelReport	Held order cancel report.
cmiIntermarketMessages.HeldOrderCancelReportStruct.cancelReqId	Identifier of the cancel request.
cmiIntermarketMessages.HeldOrderCancelReportStruct.heldOrderDetail	Specifications of the held order.
cmiIntermarketMessages.HeldOrderCancelRequestStruct.cancelReqId	Identifier of the cancel request.
cmiIntermarketMessages.HeldOrderCancelRequestStruct.cancelRequest	Cancel request of the held order.
cmiIntermarketMessages.HeldOrderDetailStruct.heldOrder	Order being held for price improvement.
cmiIntermarketMessages.HeldOrderDetailStruct.productInformation	Intermarket products information.
cmiIntermarketMessages.HeldOrderDetailStruct.statusChange	Held order status update.
cmiIntermarketMessages.HeldOrderFilledReportStruct.filledReport	The fill report of the held order.
cmiIntermarketMessages.HeldOrderFilledReportStruct.heldOrderDetail	Details of the held order that was filled.
cmiIntermarketMessages.HeldOrderStruct.currentMarketBest	Current intermarket best.
cmiIntermarketMessages.HeldOrderStruct.order	Held order information.
cmiOrder.InternalizationOrderResultStruct.matchOrderResult	Contains the information from the Order Result Struct for the internalized match order.
cmiOrder.InternalizationOrderResultStruct.primaryOrderResult	Contains the information from the Order Result Struct for the internalized primary order.
cmiOrder.InternalizationOrderResultStructV2.matchOrderResult	Contains the information from the Order Result Struct V2 for the internalized match order.
cmiOrder.InternalizationOrderResultStructV2.primaryOrderResult	Contains the information from the Order Result Struct V2 for the internalized primary order.
cmiUtil.KeyDescriptionStruct.description	Describes the key
cmiUtil.KeyDescriptionStruct.key	Generic value
cmiUtil.KeyValueStruct.key	
cmiUtil.KeyValueStruct.value	
cmiMarketData.LastSaleStructV4.classKey	Class key for the product whose current market is being reported.
cmiMarketData.LastSaleStructV4.exchange	Name of the exchange.
cmiMarketData.LastSaleStructV4.lastSalePrice	The last sale price integer form.
cmiMarketData.LastSaleStructV4.lastSaleTime	The time of the last sale (milliseconds since midnight).

cmiMarketData.LastSaleStructV4.lastSaleVolume	The volume of the last sale.
cmiMarketData.LastSaleStructV4.netPriceChange	Changes from the previous close price. For CBOE option, it is the change from the first last sale of the day. I.E. currentLastSale - yesterdayClosePrice. This is a signed value as well and can therefore be a positive, zero, or negative IntegerPrice value. Example: CurrentLastSale: 10.00 YesterdayClose: 10.05 ==> NetChange == -0.05 (represented as integer -5 with a scale of 2).
cmiMarketData.LastSaleStructV4.priceScale	The number of decimal places to use in conjunction with the integer prices.
cmiMarketData.LastSaleStructV4.productKey	Product key for the product whose current market is being reported.
cmiMarketData.LastSaleStructV4.productType	The product type (equity, option, etc.)
cmiMarketData.LastSaleStructV4.sentTime	The time the information was sent
cmiMarketData.LastSaleStructV4.tickDirection	The direction of change in current last sale price relative to the previous last sale. PLUS_TICK '+' Means the current last sale price is higher than the previous last sale price. Also known as an "uptick". MINUS_TICK '-' Means the current last sale price is lower than the previous last sale price. Also known as a "downtick". ZERO_MINUS_TICK '_' Means the current last sale price is the same as the previous last sale price, but is lower than the most recent different last sale price. ZERO_PLUS_TICK '*' Means the current last sale price is the same as the previous last sale price, but is higher than the most recent different last sale price UNKNOWN_TICK ' ' Means the tick direction field is not used or not known.
cmiMarketData.LastSaleStructV4.totalVolume	The total volume for this product.

cmiOrder.LegOrderDetailStruct.cancelledQuantity	The cancelled leg quantity.
cmiOrder.LegOrderDetailStruct.clearingFirm	The number or acronym to be associated with this leg if it is a non-option product.
cmiOrder.LegOrderDetailStruct.coverage	Whether the leg is covered or uncovered.
cmiOrder.LegOrderDetailStruct.leavesQuantity	Current remaining open quantity for the order.
cmiOrder.LegOrderDetailStruct.mustUsePrice	The optional price field when a fixed leg price is normally set to no value.
cmiOrder.LegOrderDetailStruct.originalQuantity	Original leg quantity.
cmiOrder.LegOrderDetailStruct.positionEffect	Indicates that the resulting position in the leg product is in opening or close.
cmiOrder.LegOrderDetailStruct.productKey	Details the product of the leg.
cmiOrder.LegOrderDetailStruct.side	The side of the leg.
cmiOrder.LegOrderDetailStruct.tradedQuantity	The traded leg quantity.
cmiOrder.LegOrderEntryStruct.clearingFirm	Number or acronym to be associated with this leg if it is a non-option product.
cmiOrder.LegOrderEntryStruct.coverage	Whether the leg is covered or uncovered.
cmiOrder.LegOrderEntryStruct.mustUsePrice	Optional price field when a fixed leg price is normally set to no value.
cmiOrder.LegOrderEntryStruct.positionEffect	Indicates that the resulting position in the leg product is in opening or close.
cmiOrder.LegOrderEntryStruct.productKey	The product of the leg.
cmiOrder.LegOrderEntryStructV2.extensions	Used to persist arbitrary data along with the order.
cmiOrder.LegOrderEntryStructV2.legOrderEntry	Used to enter a complex order. References the LegOrderEntryStruct.
cmiOrder.LegOrderEntryStructV2.side	Used to indicate the short sale position of a particular leg of the complex order.
cmiProduct.ListingStateStruct.productKeys	ProductKeys for the product for which the listing state has changed.
cmiProduct.ListingStateStruct.productState	New listing state of the product (see cmiConstants::ListingStates).
cmiQuote.LockNotificationStruct.buySideUserAcronyms	User's acronym on the Buy side
cmiQuote.LockNotificationStruct.classKey	Class identifier.
cmiQuote.LockNotificationStruct.extensions	
cmiQuote.LockNotificationStruct.price	Quote price.
cmiQuote.LockNotificationStruct.productKey	Product identifier.
cmiQuote.LockNotificationStruct.productType	Product type identifier.
cmiQuote.LockNotificationStruct.quantity	
cmiQuote.LockNotificationStruct.sellSideUserAcronyms	Sell side user's acronyms.
cmiQuote.LockNotificationStruct.sessionName	Trading session identifier.
cmiQuote.LockNotificationStruct.side	Side of the quote.

cmiMarketData.MarketDataDetailStruct.bestPublishedAskPrice	The published market best ask price.
cmiMarketData.MarketDataDetailStruct.bestPublishedAskVolume	The market best ask volume that is published..
cmiMarketData.MarketDataDetailStruct.bestPublishedBidPrice	The published market best bid price.
cmiMarketData.MarketDataDetailStruct.bestPublishedBidVolume	The published market best bid volume.
cmiMarketData.MarketDataDetailStruct.brokers	List of broker acronyms
cmiMarketData.MarketDataDetailStruct.contras	A list of contra acronyms.
cmiMarketData.MarketDataDetailStruct.exchangeIndicators	List of the product states of other exchanges.
cmiMarketData.MarketDataDetailStruct.extensions	Generic keys/values.
cmiMarketData.MarketDataDetailStruct.nbboAskExchanges	Exchanges with the NBBO ask price.
cmiMarketData.MarketDataDetailStruct.nbboAskPrice	The NBBO ask price.
cmiMarketData.MarketDataDetailStruct.nbboBidExchanges	Exchanges with the NBBO bid.
cmiMarketData.MarketDataDetailStruct.nbboBidPrice	The NBBO bid price.
cmiMarketData.MarketDataDetailStruct.overrideIndicator	A type of market data override used by TPF.
cmiMarketData.MarketDataDetailStruct.tradeThroughIndicator	Indicates if the last sale violated the NBBO
cmiMarketData.MarketDataHistoryDetailEntryStruct.detailData	Contains market data details.
cmiMarketData.MarketDataHistoryDetailEntryStruct.historyEntry	Market data entry details.
cmiMarketData.MarketDataHistoryDetailStruct.endTime	The end time for the market data query.
cmiMarketData.MarketDataHistoryDetailStruct.entries	List of market data entries.
cmiMarketData.MarketDataHistoryDetailStruct.isOutOfSequence	
cmiMarketData.MarketDataHistoryDetailStruct.productKeys	Product keys for the product data that is being reported.
cmiMarketData.MarketDataHistoryDetailStruct.sessionName	Name of the trading session from which the market data originated.
cmiMarketData.MarketDataHistoryDetailStruct.startTime	The start time for the market data query.
cmiMarketData.MarketDataHistoryEntryStruct.askPrice	Quote ask price
cmiMarketData.MarketDataHistoryEntryStruct.askSize	Quote with size - ask size
cmiMarketData.MarketDataHistoryEntryStruct.bidPrice	Quote bid price
cmiMarketData.MarketDataHistoryEntryStruct.bidSize	Quote with size - bid size
cmiMarketData.MarketDataHistoryEntryStruct.buyerAcronym	Buyer in the case of a price report
cmiMarketData.MarketDataHistoryEntryStruct.entryType	Type market data history entry. Refer to cmiConstants::MarketDataHistoryEntryTypes
cmiMarketData.MarketDataHistoryEntryStruct.eopType	If an expected opening price - the type of expected opening price. See cmiConstants::ExpectedOpeningPriceTypes.

cmiMarketData.MarketDataHistoryEntryStruct.exceptionCode	Exception Code
cmiMarketData.MarketDataHistoryEntryStruct.marketCondition	For market condition entries, this is the type of market condition that was reported. see cmiConstants::ProductStates.
cmiMarketData.MarketDataHistoryEntryStruct.optionalData	Additional text message. Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional. For Sweep and AIM: Enter A:AIS in the primary order to instruct CBOE to sweep all better priced protected quotes at away exchanges at the same time as the commencement of the AIM auction. For AIM Auctions: Enter A:AIM as the first characters. If a Firm does not wish to cancel the primary order when the auction expires, the Firm must enter A:AIR, instead of A:AIM, in this field. This will designate the primary order to be returned to the system and trade or book as a regular order. Options: For Customer orders, the first four characters of this field get reported to the last four characters of the CBOE Trade Match Optional Data field. These four characters get reported to the OCC.
cmiMarketData.MarketDataHistoryEntryStruct.physLocation	Physical location
cmiMarketData.MarketDataHistoryEntryStruct.prefix	Information prepended on the market data history message
cmiMarketData.MarketDataHistoryEntryStruct.price	Last sales price - if EntryType is PriceReport.
cmiMarketData.MarketDataHistoryEntryStruct.quantity	Quantity - if this is a quote entry with size or a price report.
cmiMarketData.MarketDataHistoryEntryStruct.reportTime	Time market data event was originally reported
cmiMarketData.MarketDataHistoryEntryStruct.sellerAcronym	Seller in the case of a price report
cmiMarketData.MarketDataHistoryEntryStruct.source	Market source see cmiConstants::Sources
cmiMarketData.MarketDataHistoryEntryStruct.underlyingLastSalePrice	In the case of an expected opening price - the last sales price for the underlying

cmiMarketData.MarketDataHistoryStruct.endTime	End time of the last MarketDataHistoryEntryStruct being returned.
cmiMarketData.MarketDataHistoryStruct.entries	Sequence of MarketDataHistoryEntryStructs for the product for the specified time interval being returned.
cmiMarketData.MarketDataHistoryStruct.productKeys	Product keys for the product whose market data history is being returned.
cmiMarketData.MarketDataHistoryStruct.sessionName	Name of the trading session from which the set of market data history records was generated
cmiMarketData.MarketDataHistoryStruct.startTime	Start time of the first MarketDataHistoryEntryStruct returned.
cmiMarketData.MarketVolumeStruct.multipleParties	Indicates if orders from multiple parties make up this volume
cmiMarketData.MarketVolumeStruct.quantity	Quantity for this type of contingent order. For options the quantity is expressed as contracts. For stocks the quantity is expressed as shares.
cmiMarketData.MarketVolumeStruct.volumeType	Type of volume being reported (by contingency type). See cmiConstants::VolumeTypes.
cmiMarketData.MarketVolumeStructV4.multipleParties	Indicates if orders from multiple parties make up this volume
cmiMarketData.MarketVolumeStructV4.quantity	Quantity for this type of contingent order. For options the quantity is expressed as contracts. For stocks the quantity is expressed as shares.
cmiMarketData.MarketVolumeStructV4.volumeType	Type of volume being reported (by contingency type). See cmiConstants::VolumeTypes.
cmiAdmin.MessageStruct.messageKey	Unique identifier for the message
cmiAdmin.MessageStruct.messageText	Text of the message
cmiAdmin.MessageStruct.originalMessageKey	Original message key for which this message is a response
cmiAdmin.MessageStruct.replyRequested	Indicates if a reply is requested - true if a response is requested
cmiAdmin.MessageStruct.sender	Userid of the sender of the message
cmiAdmin.MessageStruct.subject	Subject of the message
cmiAdmin.MessageStruct.timeStamp	Timestamp when message was sent
cmiMarketData.NBBOStruct.askExchangeVolume	A series of exchange names and volumes
cmiMarketData.NBBOStruct.askPrice	National best ask price
cmiMarketData.NBBOStruct.bidExchangeVolume	A series of exchange names and volumes
cmiMarketData.NBBOStruct.bidPrice	National best bid price
cmiMarketData.NBBOStruct.productKeys	Products being reported
cmiMarketData.NBBOStruct.sentTime	Time being reported
cmiMarketData.NBBOStruct.sessionName	Name of trading session from which the NBBO message originated

cmiUtil.OperationResultStruct.errorCode	Value is zero.
cmiUtil.OperationResultStruct.errorMessage	Value is an empty string.
cmiIntermarketMessages.OrderBookDetailPriceStruct.orderInfo	Sequence of order information.
cmiIntermarketMessages.OrderBookDetailPriceStruct.price	Price information.
cmiMarketData.OrderBookPriceStruct.contingencyVolume	Contingent order volume at this price for only AON and FOK volume contingencies.
cmiMarketData.OrderBookPriceStruct.price	Price in the order book.
cmiMarketData.OrderBookPriceStruct.totalVolume	The total volume at this price.
cmiMarketData.OrderBookPriceStructV2.price	Order book price.
cmiMarketData.OrderBookPriceStructV2.views	A sequence of order book price views.
cmiMarketData.OrderBookPriceViewStruct.orderBookPriceViewType	A type of order book price view.
cmiMarketData.OrderBookPriceViewStruct.viewSequence	A sequence of views.
cmiIntermarketMessages.OrderBookStruct.classKey	Product class identifier.
cmiIntermarketMessages.OrderBookStruct.contingency	Type of order contingency.

cmiIntermarketMessages.OrderBookStruct.optionalData	Optional text field. Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional. For Sweep and AIM: Enter A:AIS in the primary order to instruct CBOE to sweep all better priced protected quotes at away exchanges at the same time as the commencement of the AIM auction. For AIM Auctions: Enter A:AIM as the first characters. If a Firm does not wish to cancel the primary order when the auction expires, the Firm must enter A:AIR, instead of A:AIM, in this field. This will designate the primary order to be returned to the system and trade or book as a regular order. Options: For Customer orders, the first four characters of this field get reported to the last four characters of the CBOE Trade Match Optional Data field. These four characters get reported to the OCC.
cmiIntermarketMessages.OrderBookStruct.orderId	Identifies the order.
cmiIntermarketMessages.OrderBookStruct.orderNBBOProtectionType	Type of NBBO protection on the order.
cmiIntermarketMessages.OrderBookStruct.orderOriginType	Type of order (customer, MM, etc.)
cmiIntermarketMessages.OrderBookStruct.originalQuantity	Beginning quantity.
cmiIntermarketMessages.OrderBookStruct.price	Order price.
cmiIntermarketMessages.OrderBookStruct.productKey	Product identifier.
cmiIntermarketMessages.OrderBookStruct.receivedTime	Time the order was received.
cmiIntermarketMessages.OrderBookStruct.side	Buy or Sell side of the order.
cmiIntermarketMessages.OrderBookStruct.state	The state of the order.
cmiIntermarketMessages.OrderBookStruct.timeInForce	Time order is in force (Day, Good until cancelled, Good until date)
cmiOrder.LightOrderEntryStruct.branch	Used by the Originator to identify the Branch. Required for all roles. For W_MAIN orders, this must be a 3 character alpha only field that must be all uppercase. Orders in purely electronic sessions that do not interact with the trading floor may use between 1-3 alpha only, all uppercase characters.

cmiOrder.LightOrderEntryStruct.branchSequenceNumber	Sequence number that can be used by the Originator to uniquely identify the order. Maximum size is 4 digits and data type is numeric only. Branch + branchSequenceNumber must be unique for each day for each executingOrGiveUpFirm (give up firm or clearing firm) + correspondentFirm (if a correspondentFirm is used at all).
cmiOrder.LightOrderEntryStruct.originalQuantity	The original quantity of the light order. Required field with maximum size 9,999 for RTH and no maximum quantity for ETH. The data type is numeric only.
cmiOrder.LightOrderEntryStruct.price	Required for all roles. Limit price of the light order. Must be in minimum increments of nickels (0.05) below 3.00 and dimes (0.10) above 3.00. The whole portion of the price is maximum size of 3 in numberic format. The decimal portion is numeric only and should be given multiplied by 1 billion.
cmiOrder.LightOrderEntryStruct.productKey	ProductKey for the product for which the light order is being submitted. Required for all roles with the maximum size of 20 with data type numeric only.
cmiOrder.LightOrderEntryStruct.side	Side of the light order: either Buy or Sell. Required for all roles. The exact size is 1 and the data type is alphabetic only. This field is also used for short sale indication on simple orders. Valid values are in cmiConstants:Sides
cmiOrder.LightOrderEntryStruct.positionEffect	The effect that the light order has on the user's overall position (Opening, Closing, or Not Applicable). Required for Customer ("C") options orders. Ignored for Firm ("F") orders. This field is required for all W_MAIN orders (both single leg and strategy). However, this value must be set to a value of "N" for all orders of origin "M" or "I" in the W_MAIN session. This field is optional for orders in purely electronic sessions (ONE_MAIN, CFE_MAIN, W_STOCK) that don't interact with the trading floor. The maximum size is 1 and the data type is alpha only. This field must be set to "O" (open) for Linkage orders.
cmiOrder.LightOrderEntryStruct.coverage	Covered, uncoverd, or unspecified. Optional for all roles. The maximum size is 1 and the data value is alphabetic.
cmiOrder.LightOrderEntryStruct.isNBBOProtected	This field determines if the light order receives NBBO protection. The order has to be a customer order for this field to matter.
cmiOrder.LightOrderEntryStruct.isIOC	This field determines if the light order is an IOC order

cmiOrder.LightOrderEntryStruct.orderOriginType	Origin of the light order. See cmiConstants::OrderOrigins. Required for all roles. The maximum size is 1 character and the data type is alphabetic only. Broker-Dealer role can submit any allowable value, but Market-Maker can only enter value "M".
cmiOrder.LightOrderEntryStruct.cmtaExchange	Clearing Member Trade Agreement Exchange. Used to designate a clearing exchange.
cmiOrder.LightOrderEntryStruct.cmtaFirmNumber	Clearing Member Trade Agreement Exchange. Used to designate a clearing firm.
cmiOrder.LightOrderEntryStruct.pdpm	Preferred designated primary market maker (PDPM) for the light order
cmiOrder.LightOrderEntryStruct.userAssignedId	User Assigned order identifier.
cmiOrder.LightOrderEntryStruct.activeSession	The trading session where the light order is entered.
cmiOrder.OrderBustReinstateReportStruct.bustReinstatedReport	Details of the bust trade reinstatement.
cmiOrder.OrderBustReinstateReportStruct.reinstatedOrder	The order struct at the time of this event.
cmiOrder.OrderBustReportStruct.bustedOrder	The order struct at the time of this event.
cmiOrder.OrderBustReportStruct.bustedReport	Bust trade report
cmiOrder.OrderCancelReportStruct.cancelledOrder	The order struct at the time of this event.
cmiOrder.OrderCancelReportStruct.cancelReport	Cancel report
cmiOrder.OrderContingencyStruct.price	Price for contingency order
cmiOrder.OrderContingencyStruct.type	Order contingency type (all or none, fill or kill, etc.) (See cmiConstants::ContingencyTypes
cmiOrder.OrderContingencyStruct.volume	Quantity for this order
cmiOrder.OrderDetailStruct.orderStruct	Order record from the order handling service
cmiOrder.OrderDetailStruct.productInformation	Product name for the product for which the order was submitted
cmiOrder.OrderDetailStruct.statusChange	Status change that caused issuance of OrderDetailStruct from the CAS.

cmiOrder.OrderEntryStruct.account	Accounts must be approved by the Membership Department. Account information can be obtained by calling the getValidUser method. Users can only use one account value per product class. The exact size is 3 and the data type is alpha only. Options and Futures: Account field is required for market-maker order entry but optional for broker-dealer order entry. Accounts must be approved by the CBOE Membership Department and must be pre-configured by your API testing representative. Exact size is 3 and data type is alpha only. Linkage: For entering a Principal (P) Linkage order to an away market, this field must contain the market-maker's clearing account. This is normally the Q-Account or market-maker acronym. Clearing Information: For market-makers, this typically would be either the joint account (often called q-account) or the market-maker three-letter badge acronym. Passed through to OCC. Required for Market-Maker and DPM roles in all sessions. For Market-Maker and DPM roles, CBOE validates the value of this field on inbound orders against the CBOE Membership system. For Market-Maker and DPMs, user cannot use more than one account per class. Optional for Broker-Dealer and Firm roles. CBOE performs no validation checks on the value of this field for Broker-Dealer and Firm roles. Exact size is 3 and data type is alpha only. For futures, if desired, this field can be the same value as subaccount.
cmiOrder.OrderEntryStruct.branch	Used by the Originator to identify the Branch. Required for all roles. For W_MAIN orders, this must be a 3 character alpha only field that must be all uppercase. Orders in purely electronic sessions that do not interact with the trading floor may use between 1-3 alpha only, all uppercase characters.
cmiOrder.OrderEntryStruct.branchSequenceNumber	Sequence number that can be used by the Originator to uniquely identify the order. Maximum size is 4 digits and data type is numeric only. Branch + branchSequenceNumber must be unique for each day for each executingOrGiveUpFirm (give up firm or clearing firm) + correspondentFirm (if a correspondentFirm is used at all).

cmiOrder.OrderEntryStruct.cmta	Clearing Member Trade Agreement Firm. Used to designate a clearing firm, if different from executingOrGiveUpFirm. This field is not connected to the user ID, product class, account, or subaccount at all and is not configured by the CBOE. It can be submitted on the fly. Optional for all roles. The maximum size is 5 and data type is alphanumeric. Clearing information: The CMTA (Clearing Member Trade Agreement) field is used to designate an OCC clearing firm if it is different from the executingOrGiveUpFirm. CBOE performs no validation checks on the CMTA field against the CBOE Membership system. This field is optional for all roles in all sessions.CMTA is comprised of two components: an exchangeFirmStruct which contains the exchange code and CMTA firmNumber. If you use CMTA, then you must use submit both of these two components.The exchange string is the exchange on which your order will trade. The exchange portion of the CMTA field is alpha only.The firmNumber is the OCC clearing firm where the order will clear. The firmNumber portion of the CMTA field is numeric only. Even though the maximum size for the firmNumber component is 5, CBOE will read the first three numbers of this field to use as the OCC clearing firm. In other words, if the desired CMTA firm at the OCC is "123", do not send "00123", send "123".
cmiOrder.OrderEntryStruct.contingency	Contingency assigned to the order. CBOE currently only supports NONE, AON, FOK and IOC. The field is strongly encouraged for all orders. The maximum size is 2 with data type numberic only. For Auctions: - The OrderEntryStruct must contain an OrderContingencyStruct with type=ContingencyTypes.AUCTION_RESPONSE. The side of the auction response order must be tradable against the side indicated in the auction event.

cmiOrder.OrderEntryStruct.correspondentFirm	Optional field that is available for use by the originating firm to identify a correspondent firm. This field is not connected to the user ID, product class, account, or subaccount at all and is not configured by the CBOE. It can be submitted on the fly. The maximum size is 4 characters and the data type is uppercase alphabetic only. This field should be left blank for Linkage orders Clearing information: The correspondent firm field is used by the executing give up firm to differentiate the firm or system sending the order. The 1st three characters of this field are mapped to the optional data field on the CBOE Trade Match (CTM) record. This field has no impact on the clearing of the trade. This field is optional. CBOE performs no validation checks on the correspondentFirm field against the CBOE Membership system. Maximum size is 4 characters and data type is uppercase alpha only.
cmiOrder.OrderEntryStruct.coverage	Covered, uncoverd, or unspecified. Optional for all roles. The maximum size is 1 and the data value is alphabetic.
cmiOrder.OrderEntryStruct.cross	'C' if the order is a crossing order. Not supported in the current version of the CMi and is for futuer use in a subsequent version. CBOE currently ignores this value.

cmiOrder.OrderEntryStruct.executingOrGiveupFirm	The Member Firm for which the order will execute. Required for all roles. Broker (Broker-Dealer) role may choose from list of approved firms and the Market-Maker and DPM roles must use default firm only. The exact size is 3 and the data type is numeric only. Clearing information: This is the CBOE clearing firm that is representing the order in live trading (post trade processing firm). If no CMTA firm is present in the order, then the executingOrGiveUpFirm represents the OCC clearing firm where the order will clear. This field is required for all orders sent to CBOE for all roles in all sessions regardless of whether a CMTA firm is given or not. CBOE performs validation checks of executingOrGiveUpFirm against the CBOE Membership system on options orders routed to all sessions. Broker-Dealer and Firm roles must choose from a list of pre-approved and pre-configured executingOrGiveUpFirms and the Market-Maker and DPM roles must use default executingOrGiveUpFirm only.This field is comprised of two components: an ExchangeFirmStruct which contains the Exchange code and firmNumber.The Exchange string is the exchange on which your order will trade. The Exchange portion of the executingOrGiveUpFirm field is alpha only.If there is no CMTA given in the order, then the executingOrGiveUpFirm firmNumber will be the OCC clearing firm where the order will clear. If there is a CMTA given in the order, then the firmNumber is the CBOE clearing firm that is representing the order in live trading (post trade processing firm). The firmNumber portion of the CMTA field is numeric only. Even though the maximum size for the firmNumber component is 5, CBOE will only read the first three numbers of this field to use as the executingOrGiveUpFirm. In other words, if the desired executingOrGiveUpFirm firm is "123", do not send "00123", send "123".
cmiOrder.OrderEntryStruct.expireTime	Date Time when order is to expire. CBOE currently ignores this value.

cmiOrder.OrderEntryStruct.extensions	Used to persist arbitrary data along with the order. For Auctions: The OrderEntryStruct extensions field must contain a new field for auction response orders corresponding to the auction ID of the auction. The extensions field should contain the substring "auctionId=123:456" where 123 is the high CBOE Id and 456 is the low CBOE Id of the auction ID specified in the auction event. The standard field separator ("\u0001") must be used in between subfields of the extensions field. Valid values for multiple routing choices on Linkage orders sent through CBOE are: REDI, OES, BRASS, LIME, ASSENT

cmiOrder.OrderEntryStruct.optionalData	Optional text field. For Qualified Contingent Trade (QCT orders: Enter A:AIQ as the first characters in the primary order to instruct CBOE to treat the paired orders as QCT orders. Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional. For Sweep and AIM: Enter A:AIS in the primary order to instruct CBOE to sweep all better priced protected quotes at away exchanges at the same time as the commencement of the AIM auction. For AIM Auctions: Enter A:AIM as the first characters. If a Firm does not wish to cancel the primary order when the auction expires, the Firm must enter A:AIR, instead of A:AIM, in this field. This will designate the primary order to be returned to the system and trade or book as a regular order. Directed AIM: CBOE high and low order values prefixed by DAIM or Directed AIM match order. Orders of origin Customer ("C"): The first four characters are reported to the last four characters of CBOE Trade Match Optional Data field. These four characters are reported to OCC. Do not put "C:" in this field. Orders of origin In-Crowd Market-maker ("I") This field is not required. Do not put "I:" in this field. Futures: This field stays with the order for the life of the order. The first 16 bytes go to the OCC. This field is optional for all roles for futures orders. Maximum size is 128 characters and data type is alphanumeric. Do not send "M:" account information like is used for options clearing.

	Preferred DPM Firms that give one DPM priority in participating in a trade use this field.Firm is specified as P:firm; and can coexist with other data that may be present in this field. "Firm" is the CBOE firm acronym as listed in the Order Test Plan. Please note that the colon : and semi-colon ; are both mandatory. Linkage: This field is not used for Linkage. Strategy order: For clearing purposes, this field must be populated not the leg struct.
cmiOrder.OrderEntryStruct.orderDate	An optional order date. If not set then CBOE will default to the current date. For new orders this field should be left blank or set to the current date. Format is YYYYMMDD. Do not use hyphens "-" or slashes "/" to separate the year, month and day. The exact size is 8 and the data type is numeric only.
cmiOrder.OrderEntryStruct.orderNBBOProtectionType	This field determines if the order receives NBBO protection. The order has to be a customer order for this field to matter.
cmiOrder.OrderEntryStruct.orderOriginType	Origin of the order. See cmiConstants::OrderOrigins. Required for all roles. The maximum size is 1 character and the data type is alphabetic only. Broker-Dealer role can submit any allowable value, but Market-Maker can only enter value "M".
cmiOrder.OrderEntryStruct.originalQuantity	The original quantity of the order. Required field with maximum size 9,999 for RTH and no maximum quantity for ETH. The data type is numeric only.

cmiOrder.OrderEntryStruct.originator	Optional field. The member firm or organization that originated the order. If this field is populated, it should be populated with the User ID of the user entering the order. The maximum size is 3 and the data type is alphanumeric. Options and Futures: If a Broker_Dealer enters an order on behalf of a market-maker in the W_MAIN, CFE_MAIN, COF_MAIN or ONE_MAIN sessions, the market-maker's acronym must be entered in this field. The default exchange is CBOE, but an exchange may be specified. This is also called "Originator". Examples: "CBOE:ABC", or you may enter "ABC". Linkage: Required for entering a Principal (P) Linkage order to an away market. This field must specify the valid CBOE market maker's acronym (trading badge) of the entering market maker. Clearing information: This field would only be used for orders of origin "M", "I", and "N". This field is comprised of two components: an Exchange string which contains the Exchange code and acronym.The Exchange string is the exchange on which the market-maker will clear the trade. The Exchange portion of the originator field is alpha only.The acronym is the three-letter acronym ("badge") of the market-maker who originates the order. This field will typically be three characters (occasionally two). Orders of origin "N" Orders of origin "N" entered by a broker-dealer role must supply the three-letter acronym ("badge") of the market-maker. If a broker-dealer role submits an options order of origin "N" on behalf of a non-CBOE market-maker (e.g. a CBOE BD role enters an order on behalf of an AMEX market-maker), then the MM acronym must go into the originator portion (positions 13-15) of the optional data field (tag 9324 in FIX) and not the originator field (tag 9465 in FIX). If a broker-dealer role submits an options order of origin "N" on behalf of a non-CBOE market-maker at another exchange, and enters any value into the originator field (tag 9465 in FIX), then CBOE will reject the order. CBOE does not allow the market-maker role to enter orders of origin "N". Futures: This field is used for market-maker orders only. It is optional for OneChicago futures but not used for CFE futures.

	Orders of origin "M" Orders of origin "M" entered by a broker-dealer role must supply the three-letter acronym ("badge") of the market-maker. If a broker-dealer role submits an options order of origin "M" on behalf of a CBOE market-maker, then the three-letter (all alpha, all caps) MM acronym must go into either the originator field (tag 9465 in FIX) or the originator portion (positions 13-15) of the optional data field (tag 9324 in FIX). The market-maker role does not have to enter the originator field when entering orders of origin "M". However, if a market-maker role wishes to enter the originator field when entering orders of origin "M", then it may enter the originator acronym into either the originator field (tag 9465 in FIX) or the originator portion (positions 13-15) of the optional data field (tag 9324 in FIX). Orders of origin "I" Orders of origin "I" entered by a broker-dealer role must supply the three-letter acronym ("badge") of the market-maker. If a broker-dealer role submits an options order of origin "I" on behalf of a CBOE market-maker, then the three-letter (all alpha, all caps) MM acronym must go into the originator field (tag 9465 in FIX). If a broker-dealer role submits an options order of origin "I" on behalf of a CBOE market-maker, then it may also if it wishes put the originator acronym in the originator portion (positions 13-15) of the optional data field (tag 9324 in FIX). The market-maker role does not have to enter the originator field when entering orders of origin "I". However, if a market-maker role wishes to enter the originator field when entering orders of origin "I", then it may enter the originator acronym into either the originator field (tag 9465 in FIX) or the originator portion (positions 13-15) of the optional data field (tag 9324 in FIX).
cmiOrder.OrderEntryStruct.positionEffect	The effect that the order has on the user's overall position (Opening, Closing, or Not Applicable). Required for Customer ("C") options orders. Ignored for Firm ("F") orders. This field is required for all W_MAIN orders (both single leg and strategy). However, this value must be set to a value of "N" for all orders of origin "M" or "I" in the W_MAIN session. This field is optional for orders in purely electronic sessions (ONE_MAIN, CFE_MAIN, W_STOCK) that don't interact with the trading floor. The maximum size is 1 and the data type is alpha only. This field must be set to "O" (open) for Linkage orders.

cmiOrder.OrderEntryStruct.price	Required for all roles. Limit price of the order. Must be in minimum increments of nickels (0.05) below 3.00 and dimes (0.10) above 3.00. The whole portion of the price is maximum size of 3 in numberic format. The decimal portion is numeric only and should be given multiplied by 1 billion.
cmiOrder.OrderEntryStruct.productKey	ProductKey for the product for which the order is being submitted. Required for all roles with the maximum size of 20 with data type numeric only.
cmiOrder.OrderEntryStruct.sessionNames	Sequence of trading session names where the order is eligible to trade. At this time only one trading session name can be specified. If no trading session name is specified - the order will default to the main trading session for a product.
cmiOrder.OrderEntryStruct.side	Side of the order: either Buy or Sell. Required for all roles. The exact size is 1 and the data type is alphabetic only. This field is also used for short sale indication on simple orders. Valid values are in cmiConstants:Sides

cmiOrder.OrderEntryStruct.subaccount	The maximum size is 10 and the data type is alphanumeric. CBOE performs no validation on this field. Options and Futures: Required for all futures orders (ONE and CFE). Should not be used for orders of origin "M", "N", or "I". For entering orders of any origin other than "M", "I" or "N", (including "C" = customer), this field is optional and can contain any value. Specifies the market-maker account where the trade will clear at the OCC. This field could be the Q-account, joint account, or the market-maker acronym of the market-maker. Linkage: This field is not used for Linkage. Clearing information: CBOE performs no validation checks on subaccount against the CBOE Membership system. Maximum size is 6 and data type is alphanumeric. For Broker and Firm roles, if subaccount is used then the account field is not required. If subaccount is specified, then the trade will clear into the subaccount instead of the account field. Options: This field is optional for all roles in the W_MAIN session. Futures: Subaccount is required for CFE and OneChicago futures. It specifies the account into which the trade will clear.
cmiOrder.OrderEntryStruct.timeInForce	Time the order is in force (Day, Good until Cancelled, Good until Date) (See cmiConstants::TimesInForce). Required for all roles.
cmiOrder.OrderEntryStruct.userAssignedId	Similar to Optional Data except the information never leaves CBOEdirect. Optional for all roles. The maximum size is 20 and the data type is aphanumeric.
cmiOrder.OrderFilledReportStruct.filledOrder	The order struct at the time of this event.
cmiOrder.OrderFilledReportStruct.filledReport	Fill report details
cmiIntermarketMessages.OrderFillRejectStruct.fillRejectReports	A series of fill reject reports.
cmiIntermarketMessages.OrderFillRejectStruct.rejectedFillOrder	Rejected fill order detail.
cmiOrder.OrderIdStruct.branch	Used by the Originator to Identify the Branch. This is a 3 character alphabetic only field.
cmiOrder.OrderIdStruct.branchSequenceNumber	Unique sequence number for the order that is assigned by the Originator.This is a 4 digit numeric only field.

cmiOrder.OrderIdStruct.correspondentFirm	Optional field that is available for use by the originating firm to identify a correspondent firm.
cmiOrder.OrderIdStruct.executingOrGiveUpFirm	Member Firm Acronym of the firm for which the order will execute. This is often the same as the Originator. If the Originator is a member organization - the executing Firm will be set to the member firm that the order will be given up to. Note that CBOE Trading operations must pre-approve and configure all give up relationships.
cmiOrder.OrderIdStruct.highCboeId	Major part of CBOE order identifier
cmiOrder.OrderIdStruct.lowCboeId	Minor or least significant part of CBOE order key
cmiOrder.OrderIdStruct.orderDate	Date assigned to the order by the Order Service. It is in YYYYMMDD format.
cmiIntermarketMessages.OrderReminderStruct.reminderId	Unique reminder id.
cmiIntermarketMessages.OrderReminderStruct.reminderReason	Reason for the reminder.
cmiIntermarketMessages.OrderReminderStruct.timeSent	Time the reminder was sent.
cmiOrder.LightOrderResultStruct.branch	Used by the Originator to Identify the Branch. This is a 3 character alphabetic only field.
cmiOrder.LightOrderResultStruct.branchSequenceNumber	Unique sequence number for the order that is assigned by the Originator. This is a 4 digit numeric only field.
cmiOrder.LightOrderResultStruct.orderHighId	Major part of CBOE order identifier
cmiOrder.LightOrderResultStruct.orderLowId	Minor or least significant part of CBOE order key
cmiOrder.LightOrderResultStruct.side	The side of the light order, either Buy or Sell
cmiOrder.LightOrderResultStruct.leavesQuantity	Remaining open quantity for the light order
cmiOrder.LightOrderResultStruct.tradedQuantity	The amount of the light order that was traded
cmiOrder.LightOrderResultStruct.cancelledQuantity	The amount of the light order that was cancelled.
cmiOrder.LightOrderResultStruct.reason	The reason for the result struct
cmiOrder.LightOrderResultStruct.time	Date and time of the result stuct
cmiOrder.LightOrderReplaceResultStruct.originalOrder	Returns the LightOrderResultStruct of the original order
cmiOrder.LightOrderReplaceResultStruct.newOrder	Returns the LightOrderResultStruct of the new order
cmiOrder.OrderResultStruct.orderID	Order ID for the associated order.
cmiOrder.OrderResultStruct.result	Returns the OperationResultStruct used to define generic operation when an exception is not thrown. See cmiUtil.
cmiOrder.OrderResultStructV2.order	Returns the order information from the OrderStruct.
cmiOrder.OrderResultStructV2.result	Returns the OperationResultStruct used to define generic operation when an exception is not thrown. See cmiUtil.
cmiOrder.OrderStruct.account	The order struct at the time of this event.

cmiOrder.OrderStruct.activeSession	Indicates the session thats currently active.
cmiOrder.OrderStruct.averagePrice	Average price of all executions against this order for the entire life of the order.
cmiOrder.OrderStruct.cancelledQuantity	Quantity cancelled
cmiOrder.OrderStruct.cancelReportType	Type of report - overall strategy, leg of strategy.
cmiOrder.OrderStruct.classKey	Class Key for the product being traded
cmiOrder.OrderStruct.cmta	Clearing Member Trade Agreement Firm. Used to designate a clearing firm, if different from ExecutingOrGiveUpFirm.
cmiOrder.OrderStruct.contingency	Contingency assigned to this order.
cmiOrder.OrderStruct.coverage	Covered, uncovered, or unspecified
cmiOrder.OrderStruct.cross	'C' if the order is a crossing order.
cmiOrder.OrderStruct.crossedOrder	The order id of the crossed order - if the order is a cross. Not supported in the current version of the CMi and is for future use in a subsequent version.
cmiOrder.OrderStruct.expireTime	Date Time when order is to expire.
cmiOrder.OrderStruct.extensions	Additional information persisted with the order.
cmiOrder.OrderStruct.leavesQuantity	Remaining open quantity.
cmiOrder.OrderStruct.legOrderDetails	Leg detail of a crossing order.

cmiOrder.OrderStruct.optionalData	Optional data. Firms that want to give one DPM priority in participating in a trade use this field using P:EXCH.FIRM; or P:FIRM; EXCH is the exchange acronym. FIRM is the Firm acronym. EXCH is optional. For Sweep and AIM: Enter A:AIS in the primary order to instruct CBOE to sweep all better priced protected quotes at away exchanges at the same time as the commencement of the AIM auction. For AIM Auctions: Enter A:AIM as the first characters. If a Firm does not wish to cancel the primary order when the auction expires, the Firm must enter A:AIR, instead of A:AIM, in this field. This will designate the primary order to be returned to the system and trade or book as a regular order. Options: For Customer orders, the first four characters of this field get reported to the last four characters of the CBOE Trade Match Optional Data field. These four characters get reported to the OCC.
cmiOrder.OrderStruct.orderId	Order identification - contains both the user order identification and the system assigned OrderKey
cmiOrder.OrderStruct.orderNBBOProtectionType	This field determines if the order receives NBBO protection. The order has to be a customer order for this field to matter.
cmiOrder.OrderStruct.orderOriginType	Origin of the order. 'C ' - customer, 'F'- firm, 'B' - Broker/Dealer, 'X' - Customer Broker/Dealer, 'M' - Market Maker.
cmiOrder.OrderStruct.originalQuantity	Original quantity of the order.
cmiOrder.OrderStruct.originator	The Member Firm or Member Organization the originated the order.
cmiOrder.OrderStruct.orsId	Order Routing System Identifier
cmiOrder.OrderStruct.positionEffect	Position Effect (Opening or closing or none)
cmiOrder.OrderStruct.price	Bid or offer price - depending on side
cmiOrder.OrderStruct.productKey	ProductKey for the product for which the order is being submitted.
cmiOrder.OrderStruct.productType	CBOE product type
cmiOrder.OrderStruct.receivedTime	Time the order was received by the system
cmiOrder.OrderStruct.sessionAveragePrice	Average price of all executions against this order for this trading session.

cmiOrder.OrderStruct.sessionCancelledQuantity	Quanity cancelled for this order during the current trading session.
cmiOrder.OrderStruct.sessionNames	Names of the trading sessions where the user permits the order to be traded.
cmiOrder.OrderStruct.sessionTradedQuantity	Quantity of the order that was executed during this trading session.
cmiOrder.OrderStruct.side	Side of the order either Buy or Sell.
cmiOrder.OrderStruct.source	Source of the order - either SBT or TPF (See cmiConstants::Sources)
cmiOrder.OrderStruct.state	Order State (booked, active, cancelled, etc.) (See cmiConstants::OrderStates)
cmiOrder.OrderStruct.subaccount	Sub account
cmiOrder.OrderStruct.timeInForce	Time the order is in force (Day, Good until Cancelled, Good until Date) (See cmiConstants::TimesInForce).
cmiOrder.OrderStruct.tradedQuantity	Quantity traded
cmiOrder.OrderStruct.transactionSequenceNumber	Transaction Sequence Number to identify the ordering of order reports. OrderDetailStructs can be received via different callbacks - ordering is not guaranteed
cmiOrder.OrderStruct.userAcronym	User assigned acronym.
cmiOrder.OrderStruct.userAssignedID	User Assigned order identifier - added to track user assigned quote identifiers, can also be used when entering orders
cmiOrder.OrderStruct.userId	Trading system userid of the user that submitted the order
cmiOrder.ORDOrderStruct.bookedQuantity	The quantity that went to the order book.
cmiOrder.ORDOrderStruct.orderID	The order identification
cmiOrder.ORDOrderStruct.state	The state of the order.
cmiProduct.PendingAdjustmentStruct.active	Indicates if this adjustment is currently active
cmiProduct.PendingAdjustmentStruct.classKey	Unique identifier of the class for which the pending adjustment is being reported
cmiProduct.PendingAdjustmentStruct.effectiveDate	Effective date of adjustment.
cmiProduct.PendingAdjustmentStruct.productsPending	List of products belonging to this series that will be modified (added, deleted, updated) on the effective date. For options - these are a list of series that will be introduced when the effective date of the adjustment occurs.
cmiProduct.PendingAdjustmentStruct.submittedDate	Date adjustment was submitted
cmiProduct.PendingAdjustmentStruct.type	Type of adjustment (see cmiConstants::PriceAdjustmentTypes)
cmiProduct.PendingNameStruct.action	Action to be performed on product (see cmiConstants::PriceAdjustmentActions).
cmiProduct.PendingNameStruct.pendingNameStruct	New naming information for product (can include exercise price, symbol, etc.).
cmiProduct.PendingNameStruct.productStruct	The product to be updated (add, changed, deleted) on the effective date.
cmiOrder.PendingOrderStruct.currentOrder	The current order whose product has a pending adjustment.
cmiOrder.PendingOrderStruct.pendingOrder	The order that will replace the current order - when the pending adjustment takes effect.

cmiOrder.PendingOrderStruct.pendingProductName	Provides the name of the current product and the adjusted product.
cmiUser.PreferenceStruct.name	Name of a preference - that can include a hierarchy.
cmiUser.PreferenceStruct.value	Value of the preference.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.highOpeningPrice	High opening price in the price range used in the Stock ITS.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.lowOpeningPrice	The low opening price in the price range used in the Stock ITS.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.preOpenOriginType	Origin type used in pre-open by the Stock ITS.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.preOpenType	A pre-open message type used by the Stock ITS.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.principalQuantity	Principal order quantity in ITS.
cmiIntermarketMessages.PreOpeningIndicationPriceStruct.side	Buy or sell side of the order in ITS
cmiIntermarketMessages.PreOpeningResponsePriceStruct.agencyQuantity	Quantitiy of agency participant at the opening.
cmiIntermarketMessages.PreOpeningResponsePriceStruct.orderState	State of the order. Optional, only valid state-Cancel.
cmiIntermarketMessages.PreOpeningResponsePriceStruct.principalQuantitiy	Quantitiy of Principal participant at the opening.
cmiIntermarketMessages.PreOpeningResponsePriceStruct.responsePrice	The price the participant is responding with for the opening. See cmiUtil::PriceStruct
cmiIntermarketMessages.PreOpeningResponsePriceStruct.side	Side of the order.
cmiUtil.PriceStruct.fraction	Decimal portion to be divided by the cmiConstants::PriceScale.DEFAULT_SCALE
cmiUtil.PriceStruct.type	Type of price stored in PriceStruct (See cmiConstants::PriceTypes)
cmiUtil.PriceStruct.whole	Whole number portion of the price
cmiProduct.ProductDescriptionStruct.baseDescriptionName	The ProductDescription that this particular product description is based upon
cmiProduct.ProductDescriptionStruct.maxStrikePrice	Maximum strike price for this product.
cmiProduct.ProductDescriptionStruct.minimumAbovePremiumFraction	Fraction (tick size) to be used if the premium price is above the PremiumBreakPoint
cmiProduct.ProductDescriptionStruct.minimumBelowPremiumFraction	Fraction (tick size) to be used if the premium price is below the PremiumBreakPoint
cmiProduct.ProductDescriptionStruct.minimumStrikePriceFraction	Minimum strike price fraction for the product. This is used by the trading system when adjusting products due to corporate actions (splits).
cmiProduct.ProductDescriptionStruct.name	Name of this Product Description
cmiProduct.ProductDescriptionStruct.premiumBreakPoint	Premium (price where product is trading) break point for changes in tick size. If premium is below breakpoint the tick size is the minimumBelowPremiumFraction. If premium is above the breakpoint, then use minimumAbovePremiumFraction.

cmiProduct.ProductDescriptionStruct.premiumPriceFormat	Price format for premium - either fraction or decimal
cmiProduct.ProductDescriptionStruct.priceDisplayType	Not used in SBT
cmiProduct.ProductDescriptionStruct.strikePriceFormat	Price format (fraction or decimal) of the strike price
cmiProduct.ProductDescriptionStruct.underlyingPriceFormat	Price format of the underlying product
cmiProduct.ProductKeysStruct.classKey	Unique identifier of the class to which the product belongs
cmiProduct.ProductKeysStruct.productKey	Unique identifier for the product
cmiProduct.ProductKeysStruct.productType	Product type (see cmiConstants::ProductTypes)
cmiProduct.ProductKeysStruct.reportingClass	Reporting class of the product
cmiProduct.ProductNameStruct.exercisePrice	Exercise Price
cmiProduct.ProductNameStruct.expirationDate	Expiration date is the date that the option or future will expire. This is usually the day after the third Friday of the respective month.
cmiProduct.ProductNameStruct.optionType	Option Type
cmiProduct.ProductNameStruct.productSymbol	Symbol for product - this is the symbol of the class that is associated to the product - usually the underlying symbol.
cmiProduct.ProductNameStruct.reportingClass	Reporting class symbol. This is the symbol that is displayed for trading.
cmiSession.ProductStateStruct.productKeys	Product keys for the product whose state is being reported.
cmiSession.ProductStateStruct.productState	State of the product (open, closed, halted, etc.)
cmiSession.ProductStateStruct.productStateTransactionSequenceNumber	Transaction sequence number used to sequence product state reports across product state change messages and session product messages.
cmiSession.ProductStateStruct.sessionName	Name of the session for which the product state is applicable.
cmiProduct.ProductStruct.activationDate	Activation date
cmiProduct.ProductStruct.companyName	Company name
cmiProduct.ProductStruct.createdTime	Time product was created
cmiProduct.ProductStruct.description	Description of the product
cmiProduct.ProductStruct.inactivationDate	Inactivation date
cmiProduct.ProductStruct.lastModifiedTime	Last modification time
cmiProduct.ProductStruct.listingState	Listing state of product (see cmiConstants:: ListingStates)
cmiProduct.ProductStruct.maturityDate	Maturity date is used for bonds only. It will be empty for options, futures, or any other non-bond product.
cmiProduct.ProductStruct.opraMonthcode	OPRA month code
cmiProduct.ProductStruct.opraPriceCode	OPRA Price Code
cmiProduct.ProductStruct.productKeys	Product keys for the product
cmiProduct.ProductStruct.productName	Product name
cmiProduct.ProductStruct.standardQuantity	Standard quantity

cmiProduct.ProductStruct.unitMeasure	Unit of measure
cmiProduct.ProductTypeStruct.createdTime	Date and time when product was created.
cmiProduct.ProductTypeStruct.description	Product Description.
cmiProduct.ProductTypeStruct.lastModifiedtime	Last date time the product was modified.
cmiProduct.ProductTypeStruct.name	Product Name
cmiProduct.ProductTypeStruct.type	Product Type.
cmiUser.ProfileStruct.account	Account to be used when reporting trades against quotes for this class
cmiUser.ProfileStruct.classKey	Class key for class for which the profile is applicable
cmiUser.ProfileStruct.executingOrGiveUpFirm	Executing or give up firm to be reported on trades for quotes on this class
cmiUser.ProfileStruct.subAccount	SubAccount to be used when reporting trades that occured based upon quotes for this class
cmiQuote.QuoteBustReportStruct.bustedReport	Bust report
cmiQuote.QuoteBustReportStruct.productKeys	Product keys of the product for which the trade is being busted
cmiQuote.QuoteBustReportStruct.productName	Product name of the product for which the trade is being busted
cmiQuote.QuoteBustReportStruct.quoteKey	Identifier of the quote for which the trade is being busted
cmiQuote.QuoteBustReportStruct.statusChange	Quote status as a result of the trade bust (See cmiConstants::StatusUpdateReasons).
cmiQuote.QuoteCancelReportStruct.cancelReason	The reason for the quote cancel
cmiQuote.QuoteCancelReportStruct.productKeys	Product keys of the product for which the trade is being cancelled.
cmiQuote.QuoteCancelReportStruct.productName	Name of the product for which the trade is being cancelled.
cmiQuote.QuoteCancelReportStruct.quoteKey	Identifier of the quote for which the trade is being cancelled.
cmiQuote.QuoteCancelReportStruct.statusChange	Quote status as a result of the cancel
cmiQuote.QuoteDeleteReportStruct.deleteReason	The reason the quote was deleted.
cmiQuote.QuoteDeleteReportStruct.quote	Details of the quote that was deleted.
cmiQuote.QuoteDetailStruct.productKeys	Product Keys for the quote being returned
cmiQuote.QuoteDetailStruct.productName	Product name for the quote being returned
cmiQuote.QuoteDetailStruct.quote	Quote
cmiQuote.QuoteDetailStruct.statusChange	Reason that the quote detail message was sent.
cmiQuote.QuoteEntryStruct.askPrice	Ask price of the quote
cmiQuote.QuoteEntryStruct.askQuantity	Ask quantity
cmiQuote.QuoteEntryStruct.bidPrice	Bid price of the quote
cmiQuote.QuoteEntryStruct.bidQuantity	Bid quantity
cmiQuote.QuoteEntryStruct.productKey	Product key for the product being quoted
cmiQuote.QuoteEntryStruct.sessionName	Name of the trading session for which the quote is to be entered.

cmiQuote.QuoteEntryStruct.userAssignedID	Optional user identifier for the quote. The identifier will be carried through system and returned on a fill report - if the quote results in a trade.
cmiQuote.QuoteEntryStructV3.quoteEntry	Data required for quote entry.
cmiQuote.QuoteEntryStructV3.quoteUpdateControlId	Identification for a quote update.
cmiQuote.QuoteEntryStructV4.extensions	Used to persist arbitrary data along with the quote.
cmiQuote.QuoteEntryStructV4.quoteEntryV3	Data required for quote entry. References the QuoteEntryStructV3.
cmiQuote.QuoteEntryStructV4.sellShortIndicator	Used to indicate the short sale position on the individual quote or on each of the quotes of the mass quotes.
cmiQuote.QuoteFilledReportStruct.filledReport	Fill report for the quote
cmiQuote.QuoteFilledReportStruct.productKeys	Product keys of the product being quoted
cmiQuote.QuoteFilledReportStruct.productName	Name of the product being quoted
cmiQuote.QuoteFilledReportStruct.quoteKey	Unique identifier for the quote that was filled
cmiQuote.QuoteFilledReportStruct.statusChange	Quote status as a result of this report. Ignore this value, using instead the statusChange value in the QuoteDetailStruct.
cmiQuote.QuoteRiskManagementProfileStruct.classKey	Class that the quote risk management parameters are to be applied
cmiQuote.QuoteRiskManagementProfileStruct.quoteRiskManagementEnabled	Set to true, quote management is enabled for this class.
cmiQuote.QuoteRiskManagementProfileStruct.timeWindow	The time window for the quote risk management threshold is expressed in milliseconds.
cmiQuote.QuoteRiskManagementProfileStruct.volumeThreshold	Volume threshold for the class
cmiQuote.QuoteStruct.askPrice	Ask price of the quote
cmiQuote.QuoteStruct.askQuantity	Ask quantity of the quote
cmiQuote.QuoteStruct.bidPrice	Bid price of the quote
cmiQuote.QuoteStruct.bidQuantity	Bid quantity of the quote
cmiQuote.QuoteStruct.productKey	Unique identifier for a quote in the system
cmiQuote.QuoteStruct.quoteKey	Unique identifier for a quote that is assigned by the system
cmiQuote.QuoteStruct.sessionName	Name of the trading session for which the quote is applicable.
cmiQuote.QuoteStruct.transactionSequenceNumber	Sequence number assigned by server to guarantee uniqueness of message and ordering between CAS and Exchange services.
cmiQuote.QuoteStruct.userAssignedID	User supplied identifier. Provided for FIX compatibility.
cmiQuote.QuoteStruct.userId	Identifies the user that submitted the quote
cmiQuote.QuoteStructV3.quote	Data required for quote entry.
cmiQuote.QuoteStructV3.quoteUpdateControlId	Identification for quote update.
cmiMarketData.RecapStruct.askPrice	Ask Price

cmiMarketData.RecapStruct.askSize	Ask Size
cmiMarketData.RecapStruct.askTime	Time of the last Ask Price
cmiMarketData.RecapStruct.bidDirection	Indicates if the bid is greater or less than the previous bid '+' indicates bid is greater than previous, '-' indicates bid is lower than previous.
cmiMarketData.RecapStruct.bidPrice	Bid Price
cmiMarketData.RecapStruct.bidSize	Bid Size
cmiMarketData.RecapStruct.bidTime	Time of the last bid
cmiMarketData.RecapStruct.closePrice	Closing Price
cmiMarketData.RecapStruct.highPrice	Daily High Price
cmiMarketData.RecapStruct.isOTC	True if it's an Over The Counter
cmiMarketData.RecapStruct.lastSalePrice	Last Sales Price
cmiMarketData.RecapStruct.lastSaleVolume	Last Sales Volume
cmiMarketData.RecapStruct.lowPrice	Daily Low Price
cmiMarketData.RecapStruct.netChange	Change from the previous sale.
cmiMarketData.RecapStruct.netChangeDirection	The net change is used to indicate how the price has changed with respect to the previous closing price. '+' indicates the price has increased since previous close and '-' indicates the price has decreased since previous close.
cmiMarketData.RecapStruct.openInterest	Open Interest is not supported by CBOEdirect or OneChicago and can be obtained from the Options Clearing Corporation (OCC) or CBOE Financial Network (CFN).
cmiMarketData.RecapStruct.openPrice	Opening Price
cmiMarketData.RecapStruct.previousClosePrice	Previous Closing Price
cmiMarketData.RecapStruct.productInformation	Product name for the product whose recap is being reported
cmiMarketData.RecapStruct.productKeys	Product keys for the product whose recap is being reported
cmiMarketData.RecapStruct.recapPrefix	Prefix information provided from external market data sources
cmiMarketData.RecapStruct.sessionName	Name of trading session from which the market recap originated
cmiMarketData.RecapStruct.tick	Tick size for product
cmiMarketData.RecapStruct.tickDirection	Direction from the previous sale. '+' indicates if this price has increased, '-' indicates the price has decreased.
cmiMarketData.RecapStruct.totalVolume	Daily Total Volume
cmiMarketData.RecapStruct.tradeTime	Time of the last trade
cmiMarketData.RecapStructV4.classKey	Class key for the product whose current market is being reported.
cmiMarketData.RecapStructV4.exchange	Name of the exchange.
cmiMarketData.RecapStructV4.highPrice	Today's high price.
cmiMarketData.RecapStructV4.lowPrice	Today's low price.

cmiMarketData.RecapStructV4.openPrice	Today's opening price.
cmiMarketData.RecapStructV4.previousClosePrice	Previous closing price.
cmiMarketData.RecapStructV4.priceScale	The number of decimal places to use in conjunction with the integer prices.
cmiMarketData.RecapStructV4.productKey	Product key for the product whose current market is being reported.
cmiMarketData.RecapStructV4.productType	The product type (equity, option, etc.)
cmiMarketData.RecapStructV4.sentTime	The time the information was sent.
cmiMarketData.RecapStructV4.statusCodes	Currently only used for equities, not options. Indicates whether there is Dow Jones News, Reuters News, whether the stock is Ex-Dividend, and whether it's opened or closed.
cmiProduct.ReportingClassStruct.activationDate	The date the reporting class was activated.
cmiProduct.ReportingClassStruct.classKey	Unique identifier for the reporting class.
cmiProduct.ReportingClassStruct.contractSize	The size of the contract traded for the reporting class.
cmiProduct.ReportingClassStruct.createdTime	The time the reporting class was created.
cmiProduct.ReportingClassStruct.inactivationDate	The date the reporting class was inactivated.
cmiProduct.ReportingClassStruct.lastModifiedTime	The last time the reporting class was modified.
cmiProduct.ReportingClassStruct.listingState	The list state for the reporting class.
cmiProduct.ReportingClassStruct.productClassKey	Identifier for the product class.
cmiProduct.ReportingClassStruct.productClassSymbol	Ticker symbol for the product class.
cmiProduct.ReportingClassStruct.productType	Identifies the product type for the associated reporting class.
cmiProduct.ReportingClassStruct.reportingClassSymbol	Ticker symbol for the reporting class.
cmiProduct.ReportingClassStruct.transactionFeeCode	Identifes the associated fee for the trade.
cmiQuote.RFQEntryStruct.productKey	Unique identifier of the product for which an RFQ is requested
cmiQuote.RFQEntryStruct.quantity	Quantity of the order
cmiQuote.RFQEntryStruct.sessionName	Name of trading session for which the quote is being requested.
cmiQuote.RFQStruct.entryTime	Time the RFQ was entered
cmiQuote.RFQStruct.productKeys	Product keys for the product for which an RFQ is requested
cmiQuote.RFQStruct.quantity	Quantity for which the quote is requested
cmiQuote.RFQStruct.rfqType	Indicates if the RFQ was generated manually or by an automated quotation system
cmiQuote.RFQStruct.sessionName	Name of the trading session for which the request for quote has been requested.
cmiQuote.RFQStruct.timeToLive	Time the RFQ is valid. This is the response window assigned by the exchange.
cmiIntermarketMessages.SatisfactionAlertStruct.alertHdr	
cmiIntermarketMessages.SatisfactionAlertStruct.extensions	
cmiIntermarketMessages.SatisfactionAlertStruct.lastSale	Last sale information. See cmiMarketData::TickerStruct
cmiIntermarketMessages.SatisfactionAlertStruct.side	The side (buy or sell) of the S order that was traded.

cmiIntermarketMessages.SatisfactionAlertStruct.tradedThroughOrders	Orders that were traded through.
cmiIntermarketMessages.SatisfactionAlertStruct.tradedThroughPrice	The trade through price.
cmiIntermarketMessages.SatisfactionAlertStruct.tradedThroughquantity	The trade through quantity.
cmiSession.SessionClassDetailStruct.classDetail	Session class struct for a class that trades on a specific trading session.
cmiSession.SessionClassDetailStruct.products	The list of products for the class that trades on a specific trading session. It is important to note that the system is capable of having different products eligible for different trading sessions.
cmiSession.SessionClassStruct.classState	NOT IMPLEMENTED AT THIS TIME.
cmiSession.SessionClassStruct.classStateTransactionSequenceNumber	Transaction sequence number used for sequencing class state reports across SessionClassStruct and ClassStateStruct messages.
cmiSession.SessionClassStruct.classStruct	Attributes of the class
cmiSession.SessionClassStruct.eligibleSessions	List of all trading sessions for which this class is eligible for trading.
cmiSession.SessionClassStruct.sessionName	Name of the trading session which the class trades upon.
cmiSession.SessionClassStruct.underlyingSessionName	Name of the session where the underlying trades.
cmiSession.SessionManagerStructV2.sessionManager	
cmiSession.SessionManagerStructV2.sessionManagerV2	
cmiSession.SessionProductStruct.productState	State of the product (open, closed, halted, etc.)
cmiSession.SessionProductStruct.productStateTransactionSequenceNumber	Transaction sequence number used to sequence product state reports in a client application across SessionProductStruct and ProductStateStructs.
cmiSession.SessionProductStruct.productStruct	Attributes of the product
cmiSession.SessionProductStruct.sessionName	Name of the trading session upon which the product trades. This is the trading session that was queried and also the trading session for which the product state is applicable.
cmiUser.SessionProfileStruct.account	The session account.
cmiUser.SessionProfileStruct.classKey	The class identifier for the session.
cmiUser.SessionProfileStruct.executingGiveupFirm	The executing give up firm for the session.
cmiUser.SessionProfileStruct.isAccountBlanked	Account is left blank.
cmiUser.SessionProfileStruct.originCode	
cmiUser.SessionProfileStruct.sessionName	Profile session name.
cmiUser.SessionProfileStruct.subAccount	Subaccount for the session.
cmiUser.SessionProfileUserStruct.accounts	List of accounts assigned to this user for the session.
cmiUser.SessionProfileUserStruct.assignedClasses	List of assigned classes for this user for the session.
cmiUser.SessionProfileUserStruct.defaultProfile	Generic profile for all classes and all sessions.
cmiUser.SessionProfileUserStruct.defaultSessionProfiles	Lists session default profiles.

cmiUser.SessionProfileUserStruct.dpms	List of DPMs assigned to this user for the session.
cmiUser.SessionProfileUserStruct.executingGiveupFirms	List of executing or give up firms for this user for the session.
cmiUser.SessionProfileUserStruct.firm	The user's firm for the session.
cmiUser.SessionProfileUserStruct.fullName	The user's full name.
cmiUser.SessionProfileUserStruct.role	The user's role for the session.
cmiUser.SessionProfileUserStruct.sessionProfilesByClass	List of session profiles by class, excluding all profiles with default class keys.
cmiUser.SessionProfileUserStruct.userAcronym	The user acronym for the session.
cmiUser.SessionProfileUserStruct.userId	The user ID for the session--unique per user.
cmiSession.SessionStrategyLegStruct.productKey	Identifier of the product.
cmiSession.SessionStrategyLegStruct.ratioQuantity	The quanity of this product that is required to make up the overall strategy.
cmiSession.SessionStrategyLegStruct.sessionName	The trading session name where the leg is being traded.
cmiSession.SessionStrategyLegStruct.side	The side of the leg product.
cmiSession.SessionStrategyStruct.sessionProductStruct	Attributes of the product.
cmiSession.SessionStrategyStruct.sessionStrategyLegs	Individual products that make up the strategy.
cmiSession.SessionStrategyStruct.strategyType	The type of strategy eligible for trading in a trading session.
cmiStrategy.StrategyLegStruct.product	Product to be included as a leg of a strategy product
cmiStrategy.StrategyLegStruct.ratioQuantity	Ratio of this product to the other products in the strategy
cmiStrategy.StrategyLegStruct.side	Is the product to be bought or sold as part of this strategy product
cmiStrategy.StrategyRequestStruct.strategyLegs	Sequence of strategy legs - with one product per leg
cmiStrategy.StrategyStruct.product	Product information for the Strategy product
cmiStrategy.StrategyStruct.strategyLegs	Sequence of strategy legs - one for each product that makes up the strategy product
cmiStrategy.StrategyStruct.strategyType	Defines the type of strategy (spread, combination, etc).
cmiMarketData.TickerStruct.exchangeSymbol	Product Symbol as reported by the exchange that is the source of the ticker data
cmiMarketData.TickerStruct.lastSalePrice	Last Sales Price
cmiMarketData.TickerStruct.lastSaleVolume	Last Sales Volume
cmiMarketData.TickerStruct.productKeys	Product keys for the product whose ticker data is being reported
cmiMarketData.TickerStruct.salePostfix	Suffix information that is appended to the end of the ticker information by the reporting market
cmiMarketData.TickerStruct.salePrefix	Prefix information provided by the market reporting the sale. OPNT (Opening Trade) is used if the first trade is part of the opening rotation. FTAO (First Trade After Opening) is used if the first trade occurs after the opening. CNCO (Cancel Opening) is a legitimate code for OPRA and is issued for a bust that affects the opening trade.
cmiMarketData.TickerStruct.sessionName	Name of the trading session from which this last sale data originated

cmiMarketData.TickerStructV4.classKey	Class key for the product whose current market is being reported.
cmiMarketData.TickerStructV4.exchange	Name of the exchange.
cmiMarketData.TickerStructV4.priceScale	The number of decimal places to use in conjunction with the integer prices.
cmiMarketData.TickerStructV4.productKey	Product key for the product whose current market is being reported.
cmiMarketData.TickerStructV4.productType	The product type (equity, option, etc.)
cmiMarketData.TickerStructV4.salePostfix	Suffix information that is appended to the end of the ticker information by the reporting market. See the api-08, the Market Data Express document, for suffix definitions.
cmiMarketData.TickerStructV4.salePrefix	Prefix information provided by the market reporting the sale. A dot '.' will be used in front of the prefixes-- .OPNT (Opening Trade) is used if the first trade is part of the opening rotation. .FTAO (First Trade After Opening) is used if the first trade occurs after the opening. .CNCO (Cancel Opening) is a legitimate code for OPRA and is issued for a bust that affectes the opening trade.
cmiMarketData.TickerStructV4.sentTime	The time the information was sent.
cmiMarketData.TickerStructV4.tradePrice	The trade price.
cmiMarketData.TickerStructV4.tradeTime	The time of the trade (milliseconds since midnight).
cmiMarketData.TickerStructV4.tradeVolume	The volume for this trade.
cmiUtil.TimeStruct.fraction	Value between 0 and the denominator-1
cmiUtil.TimeStruct.hour	value between 0 and 24
cmiUtil.TimeStruct.minute	value between 0 and 60
cmiUtil.TimeStruct.second	value between 0 and 60
cmiSession.TradingSessionStateStruct.sessionName	Name of the trading session for which the state change is being reported.
cmiSession.TradingSessionStateStruct.sessionState	New Trading Session State.
cmiSession.TradingSessionStruct.endTime	End of session time.
cmiSession.TradingSessionStruct.sequenceNumber	Sequence number used to guarantee uniqueness and order of message between CAS and Exchange Services.
cmiSession.TradingSessionStruct.sessionName	Name of the trading session.
cmiSession.TradingSessionStruct.startTime	Starting time of trading session.
cmiSession.TradingSessionStruct.state	Trading Session State (See cmiConstants::TradingSessionStates).

cmiUser.UserLogonStruct.loginMode	The loginMode is of type LoginSessionMode. There are three modes which you can login to the CAS. The LoginSessionModes interface provides the different types of login modes as constants. The first is Stand-Alone testing - which refers to a client application program communicating with the CAS Simulator. Network Testing mode refers to talking to a CBOE test system over the CBOE network. Production refers to communicating to the CBOE production trading system.
cmiUser.UserLogonStruct.password	Alphanumeric string with a maximum length of 12 characters. The password is initially specified by CBOE Trading Operations. An interface is provided on the UserSessionManager object to change the password.
cmiUser.UserLogonStruct.userId	User identifier assigned by the exchange to represent a user of the trading system. The userId is unique per user.
cmiUser.UserLogonStruct.userName	Name of the user.
cmiUser.UserLogonStruct.version	The version is obtained by implementing the Version Interface in your application.
cmiQuote.UserQuoteRiskManagementProfileStruct.defaultQuoteRiskProfile	Default risk quote management profile
cmiQuote.UserQuoteRiskManagementProfileStruct.globalQuoteRiskManagementEnabled	Indicates if use is using quote risk management
cmiQuote.UserQuoteRiskManagementProfileStruct.quoteRiskProfiles	Sequence of quote risk management profiles for classes beign quoted by the user
cmiUser.UserStruct.accounts	List of accounts in use by this user
cmiUser.UserStruct.assignedClasses	Sequence of classes assigned to this user
cmiUser.UserStruct.defaultProfile	Default profile for any classes not explicitly listed in profilesByClass
cmiUser.UserStruct.dpms	
cmiUser.UserStruct.executingGiveupFirms	List of executing or give up firms for this user
cmiUser.UserStruct.firm	The member firm or organization to which the user belongs.
cmiUser.UserStruct.fullName	Full name of the user.
cmiUser.UserStruct.profilesByClass	Accounting profiles for classes that are quoted by this user
cmiUser.UserStruct.role	The role of the user (See cmiConstants::UserRoles).
cmiUser.UserStruct.userAcronym	The acronym of a user within the system.
cmiUser.UserStruct.userId	User identifier assigned by the exchange to represent a user of the trading system. The userId is unique per user.
cmiUser.UserStruct.userName	The alphanumeric identifier assigned to the user.

Constants

ActivityFieldTypes.ACCOUNT	Initial Value = 2
ActivityFieldTypes.ASK_OTY	Initial Value = 4
ActivityFieldTypes.ASK_PRICE	Initial Value = 3
ActivityFieldTypes.BID_OTY	Initial Value = 6
ActivityFieldTypes.BID_PRICE	Initial Value = 5
ActivityFieldTypes.BOOKED_QUANTITY	Initial Value = 32
ActivityFieldTypes.BUSTED_QUANTITY	Initial Value = 7
ActivityFieldTypes.CANCEL_REASON	Initial Value = 31
ActivityFieldTypes.CANCELLED_QUANTITY	Initial Value = 8
ActivityFieldTypes.CMTA	Initial Value = 9
ActivityFieldTypes.CONTINGENCY_TYPE	Initial Value = 10
ActivityFieldTypes.EVENT_STATUS	Initial Value = 11
ActivityFieldTypes.EXEC_BROKER	Initial Value = 35
ActivityFieldTypes.LEAVES_QUANTITY	Initial Value = 12
ActivityFieldTypes.MISMATCHED_QUANTITY	Initial Value = 13
ActivityFieldTypes.OPTIONAL_DATA	Initial Value = 14

ActivityFieldTypes.ORDER_STATE	Initial Value = 33
ActivityFieldTypes.ORDERID	Initial Value = 1
ActivityFieldTypes.ORIGINAL_QUANTITY	Initial Value = 15
ActivityFieldTypes.PRICE	Initial Value = 16
ActivityFieldTypes.PRODUCT	Initial Value = 34
ActivityFieldTypes.PRODUCT_STATE	Initial Value = 17
ActivityFieldTypes.QUANTITY	Initial Value = 18
ActivityFieldTypes.QUOTE_UPDATE_CONTROL_ID	Initial Value = 36 Identification for the quote update.
ActivityFieldTypes.QUOTEKEY	Initial Value = 19
ActivityFieldTypes.REINSTATED_QUANTITY	Initial Value = 20
ActivityFieldTypes.REPLACE_ORDERID	Initial Value = 21
ActivityFieldTypes.RFQ_TYPE	Initial Value = 22
ActivityFieldTypes.SIDE	Initial Value = 23
ActivityFieldTypes.TIME_IN_FORCE	Initial Value = 24
ActivityFieldTypes.TIME_TO_LIVE	Initial Value = 25
ActivityFieldTypes.TLC_QUANTITY	Initial Value = 26
ActivityFieldTypes.TRADED_QUANTITY	Initial Value = 27

ActivityFieldTypes.TRADEID	Initial Value = 28
ActivityFieldTypes.TRANSACTION_SEQUENCE_NUMBER	Initial Value = 29
ActivityFieldTypes.USER_ASSIGNED_ID	Initial Value = 30
ActivityReasons.AWAY_EXCHANGE_CANCEL	Initial Value = 199 Used for Linkage when cancel reason is not provided (could be user cancel or cancel remaining).
ActivityReasons.BROKER_OPTION	Initial Value = 100 Broker option (admission of non-compliance).
ActivityReasons.CANCEL_ON_FALLBACK	Initial Value = 800 Force cancel orders due to system fallback.
ActivityReasons.CANCEL_ON_RSS	Initial Value = 24 Cancel report reason used when a series becomes restricted. Any resting orders that would open a position will be canceled by the system.
ActivityReasons.CANCEL_PENDING	Initial Value = 101 Used for Linkage.
ActivityReasons.COMM_DELAYS	Initial Value = 126 Communication delays to OPRA.
ActivityReasons.CROSS_IN_PROGRESS	Initial Value = 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.
ActivityReasons.CROWD_TRADE	Initial Value = 102 Used for Linkage.
ActivityReasons.DUPLICATE_ORDER	Initial Value = 103 Duplicate order (e.g. dupe of CIOrdID).
ActivityReasons.EXCHANGE_CLOSED	Initial Value = 104 Exchange closed (an executing participant's order handling system receives an order when the market is closed).
ActivityReasons.FAILOVER	Initial Value = 11 This code may be used to indicate that a quote has been canceled due to CBOEdirect server failover.

ActivityReasons.GATE_VIOLATION	Initial Value = 105 Order received too soon (does not meet gate requirements)
ActivityReasons.INSUFFICIENT_CROSS_ORDER_DOLLAR_AMOUNT	Initial Value = 21 The Cross order price was less than the defined minimum dollar amount.
ActivityReasons.INSUFFICIENT_CROSS_ORDER_SIZE	Initial Value = 20 The Cross order size was less than the defined minimum order size.
ActivityReasons.INSUFFICIENT_CUSTOMER_ORDER_QUANTITY	Initial Value = 19 The Cross order quantity was less than the customer order quanitity at CBOE's best market.
ActivityReasons.INSUFFICIENT_QUANTITY	Initial Value = 5 Cancel reason that applies to quotes only.
ActivityReasons.INSUFFICIENT_QUANTITY_BUY_SIDE	Initial Value = 8
ActivityReasons.INSUFFICIENT_QUANTITY_SELL_SIDE	Initial Value = 9
ActivityReasons.INVALID_ACCOUNT	Initial Value = 106 Clearing account missing.
ActivityReasons.INVALID_AUTOEX_VALUE	Initial Value = 107 Invalid Auto-Ex value.
ActivityReasons.INVALID_CMTA	Initial Value = 108 Unknown clearing firm.
ActivityReasons.INVALID_EXCHANGE	Initial Value = 138 Currently used for TPF Linkage.
ActivityReasons.INVALID_FIRM	Initial Value = 109 Exec Broker missing.
ActivityReasons.INVALID_NBBO	Initial Value = 16 Signifies that the NBBO was invalid for the order.
ActivityReasons.INVALID_ORIGIN_TYPE	Initial Value = 110 OrderCapacity missing/invalid.
ActivityReasons.INVALID_POSITION_EFFECT	Initial Value = 111 OpenClose missing/invalid
ActivityReasons.INVALID_PRICE	Initial Value = 112 Reference price is out of bounds.

ActivityReasons.INVALID_PRODUCT	Initial Value = 113 Unknown, invalid, or ineligible (for Linkage) symbol (series, expiration, strike).
ActivityReasons.INVALID_PRODUCT_TYPE	Initial Value = 114 Complex order.
ActivityReasons.INVALID_QUANTITY	Initial Value = 115 Order exceeds limit.
ActivityReasons.INVALID_SESSION_ID	Initial Value = 13 The session is not valid.
ActivityReasons.INVALID_SIDE	Initial Value = 116 Invalid side.
ActivityReasons.INVALID_SUBACCOUNT	Initial Value = 117 Sub-account ID missing.
ActivityReasons.INVALID_TIME_IN_FORCE	Initial Value = 118 TimeInForce missing/invalid
ActivityReasons.INVALID_USER	Initial Value = 119 Account missing.
ActivityReasons.LATE_PRINT	Initial Value = 120 Late print to OPRA Tape.
ActivityReasons.LINKAGE_CONDITIONAL_FIELD_MISSING	Initial Value = 900 BusinessRejectReason - Conditionally required field missing.
ActivityReasons.LINKAGE_EXCHANGE_UNAVAILABLE	Initial Value = 901 BusinessRejectReason - Application not available (the session between the Hub and the distination participant is unavailable or the destination exchange's order handling application is down.)
ActivityReasons.LINKAGE_INVALID_DESTINATION	Initial Value = 903 BusinessRejectReason - Unknown ID (invalid DeliverToCompID)
ActivityReasons.LINKAGE_INVALID_MESSAGE	Initial Value = 902 BusinessRejectReason - Bad message.
ActivityReasons.LINKAGE_INVALID_PRODUCT	Initial Value = 904 BusinessRejectReason - Unknown Security
ActivityReasons.LINKAGE_SESSION_REJECT	Initial Value = 905 BusinessRejectReason - Other (used if a message relayed by the Hub received a session-level Reject from the destination participant.)

ActivityReasons.WASH_TRADE_PREVENTION	Initial Value = 906 Cancel reason if order is about to trade against another order or quote from the same user.
ActivityReasons.MISMATCHED_QUANTITY	Initial Value = 907 Cancel reason for mismatched quantity in cancel/replace for Light orders.
ActivityReasons.LOST_CONNECTION	Initial Value = 4 Cancel reason that applies to quotes only.
ActivityReasons.MISSING_EXEC_INFO	Initial Value = 122 Execution information missing.
ActivityReasons.NO_MATCHING_ORDER	Initial Value = 123 Used for Linkage.
ActivityReasons.NO_USER_ACTIVITY	Initial Value = 23 No quoting activity has occurred in a certain period of time.
ActivityReasons.NON_BLOCK_TRADE	Initial Value = 124 Used for Linkage.
ActivityReasons.NOT_ACCEPTED	Initial Value = 140 Currently used for TPF Linkage.
ActivityReasons.NOT_FIRM	Initial Value = 121 Instrument state is invalid for Linkage (the receiver is in Non-Firm mode).
ActivityReasons.NOT_NBBO	Initial Value = 125 Not at NBBO
ActivityReasons.NOT_WITHIN_NBBO	Initial Value = 17 The Cross order price was not within the NBBO.
ActivityReasons.NOTHING_DONE	Initial Value = 1
ActivityReasons.ORDER_TIMEOUT	Initial Value = 141 Used for Linkage.
ActivityReasons.ORDER_TOO_LATE	Initial Value = 134 Too late to enter.
ActivityReasons.ORIGINAL_ORDER_REJECTED	Initial Value = 127 Used for Linkage.
ActivityReasons.OTHER	Initial Value = 128 Used for Linkage.

ActivityReasons.PROCESSING_PROBLEMS	Initial Value = 129 Used for Linkage.
ActivityReasons.PRODUCT_HALTED	Initial Value = 130 Instrument state is invalid for Linkage (the receiver has the instrument halted)
ActivityReasons.PRODUCT_IN_ROTATION	Initial Value = 131 Instrument state is invalid for Linkage (the receiver is in Rotation)
ActivityReasons.PRODUCT_SUSPENDED	Initial Value = 172 This code may be used to indicate that a quote has been canceled due to product state transition to ProductStates.SUSPENDED
ActivityReasons.QRM_REMOVED	Initial Value = 7 Cancel reason that applies to quotes only.
ActivityReasons.QUOTE_IN_TRIGGER	Initial Value = 12 Quote is in a trigger status.
ActivityReasons.QUOTE_UPDATE_CONTROL	Initial Value = 10 Ability to update a quote.
ActivityReasons.QUOTE_UPDATES_REQUESTED	Initial Value = 25 Request for user to re-enter quotes.
ActivityReasons.REJECTED_LINKAGE_TRADE	Initial Value = 170 The Linkage trade was rejected.
ActivityReasons.SAL_IN_PROGRESS	Initial Value = 14 This activity reason is used when SAL auction is in progress.
ActivityReasons.SATISFACTION_ORD_REJ_OTHER	Initial Value = 171 Satisfaction order was rejected. Used for Linkage.
ActivityReasons.SELL_SHORT_RULE_VIOLATION	Initial Value = 22 The order violated the Sell Short rule.
ActivityReasons.SPECIAL_ADJUSTMENT	Initial Value = 6 Orders automatically canceled due to a special cash adjustment (product adjustment)
ActivityReasons.STALE_EXECUTION	Initial Value = 132 Used for Linkage.
ActivityReasons.STALE_ORDER	Initial Value = 133 Stale order (Time-to-live of received order has expired)
ActivityReasons.SYSTEM	Initial Value = 3 A non-user initiated cancel, e.g. Replacement order canceled due to the system's inability to cancel the original order.

ActivityReasons.TRADE_BUSTED	Initial Value = 135 Trade busted/corrected.
ActivityReasons.TRADE_REJECTED	Initial Value = 136 Trade rejected.
ActivityReasons.TRADE_THROUGH_CBOE	Initial Value = 18 The order price traded through CBOE's market.
ActivityReasons.TRANSACTION_FAILED	Initial Value = 139 Currently used for TPF Linkage.
ActivityReasons.UNKNOWN_ORDER	Initial Value = 137 Unknown order.
ActivityReasons.USER	Initial Value = 2 A user initiated cancel.
ActivityTypes.AWAY_EXCHANGE_MARKET	Initial Value = 315
ActivityTypes.BOOK_ORDER	Initial Value = 8
ActivityTypes.BUST_ORDER_FILL	Initial Value = 4
ActivityTypes.BUST_QUOTE_FILL	Initial Value = 107
ActivityTypes.BUST_QUOTE_LEG_FILL	Initial Value = 157
ActivityTypes.BUST_REINSTATE_ORDER	Initial Value = 5
ActivityTypes.BUST_REINSTATE_STRATEGY_LEG	Initial Value = 55
ActivityTypes.BUST_STRATEGY_LEG_FILL	Initial Value = 54
ActivityTypes.CANCEL_ALL_ORDERS	Initial Value = 11
ActivityTypes.CANCEL_ALL_QUOTES	Initial Value = 104
ActivityTypes.CANCEL_ORDER	Initial Value = 3

ActivityTypes.CANCEL_ORDER_REQUEST	Initial Value = 303 Activity type for Linkage.
ActivityTypes.CANCEL_ORDER_REQUEST_REJECT	Initial Value = 304 Activity type for Linkage.
ActivityTypes.CANCEL_ORDER_REQUEST_REJECT_REJECTED	Initial Value = 308 Activity type for Linkage.
ActivityTypes.CANCEL_QUOTE	Initial Value = 103
ActivityTypes.CANCEL_REPLACE_ORDER	Initial Value = 6
ActivityTypes.CANCEL_REPLACE_ORDER_REQUEST	Initial Value = 13
ActivityTypes.CANCEL_REPORT_REJECT	Initial Value = 305 Activity type for Linkage.
ActivityTypes.CANCEL_REPORT_REJECT_REJECTED	Initial Value = 309 Activity type for Linkage.
ActivityTypes.CANCEL_STRATEGY_LEG	Initial Value = 53
ActivityTypes.FILL_ORDER	Initial Value = 2
ActivityTypes.FILL_QUOTE	Initial Value = 102
ActivityTypes.FILL_REJECT	Initial Value = 302 Activity type for Linkage.
ActivityTypes.FILL_REJECT_REJECTED	Initial Value = 307 Activity type for Linkage.
ActivityTypes.FILL_STRATEGY_LEG	Initial Value = 52
ActivityTypes.HELD_FOR_IPP_PROTECTION	Initial Value = 12
ActivityTypes.LINKAGE_DISQUALIFIED_EXCHANGE	Initial Value = 316
ActivityTypes.MANUAL_ORDER_SR	Initial Value = 718

ActivityTypes.MANUAL_ORDER_SR_TIMEOUT	Initial Value = 719
ActivityTypes.MANUAL_ORDER_SR_TIMEOUT_FAILURE	Initial Value = 816
ActivityTypes.MANUAL_ORDER_FR	Initial Value = 720
ActivityTypes.MANUAL_ORDER_FR_TIMEOUT	Initial Value = 721
ActivityTypes.MANUAL_ORDER_FR_TIMEOUT_FAILURE	Initial Value = 722
ActivityTypes.NEW_ORDER	Initial Value = 1
ActivityTypes.NEW_ORDER_REJECT	Initial Value = 301 Activity type for Linkage.
ActivityTypes.NEW_ORDER_REJECT_REJECTED	Initial Value = 306 Activity type for Linkage.
ActivityTypes.NEW_ORDER_STRATEGY_LEG	Initial Value = 51
ActivityTypes.NEW_QUOTE	Initial Value = 101
ActivityTypes.NEW_RFQ	Initial Value = 201
ActivityTypes.PRICE_ADJUST_ORDER	Initial Value = 10
ActivityTypes.PRICE_ADJUST_ORDER_LEG	Initial Value = 60
ActivityTypes.QUOTE_LEG_FILL	Initial Value = 152
ActivityTypes.ROUTE_TO_AWAY_EXCHANGE	Initial Value = 310
ActivityTypes.STATE_CHANGE_ORDER	Initial Value = 9
ActivityTypes.SYSTEM_CANCEL_QUOTE	Initial Value = 105

ActivityTypes.UPDATE_ORDER	Initial Value = 7
ActivityTypes.UPDATE_QUOTE	Initial Value = 106
ActivityTypes.UPDATE_STRATEGY_LEG	Initial Value = 57
AlertResolutions.ADJUSTED	Initial Value = "A"
AlertResolutions.AWAY_MARKET_REFUSE_TO_TRADE_OR_FADE	Initial Value = "NU"
AlertResolutions.AWAY_MARKET_UNAVAIL_TO_TRADE	Initial Value = "IN"
AlertResolutions.BOOK_TAKEN_OUT_AFTER_NOTIFICATION	Initial Value = "BA"
AlertResolutions.CBOE_SYSTEM_PROBLEMS	Initial Value = "SP"
AlertResolutions.CONTRA_UNAVAILABLE	Initial Value = "CU"
AlertResolutions.DELAYED_REPORT	Initial Value = "DR"
AlertResolutions.ERRONEOUS_REPORT	Initial Value = "ER"
AlertResolutions.EXECUTED_UNDER_FIRM_INSTRUCTIONS	Initial Value = "FI"
AlertResolutions.FAST_MARKET_AWAY	Initial Value = "FM"
AlertResolutions.FIRM_DISCRETION	Initial Value = "FD"
AlertResolutions.FLASH_QUOTE	Initial Value = "FQ"
AlertResolutions.NBBO_FADE	Initial Value = "NF"
AlertResolutions.NBBO_LOCKED_W_CBOE	Initial Value = "LB"

AlertResolutions.NOT_ADJUSTED	Initial Value = "UA"
AlertResolutions.NOT_RESOLVED	Initial Value = "NR"
AlertResolutions.OTHER	Initial Value = "O"
AlertResolutions.PARTIAL_PRICE_ADJUSTMENT	Initial Value = "AP"
AlertResolutions.PARTIAL_QUANTITY_ADJ	Initial Value = "AQ"
AlertResolutions.POST_TRADE_QUOTE	Initial Value = "PQ"
AlertResolutions.SHUT_OFF_ERROR	Initial Value = "SE"
AlertResolutions.SINGLE_LISTED_OPTION	Initial Value = "SL"
AlertResolutions.TRADE_BUSTED	Initial Value = "TB"
AlertResolutions.TRADE_ENTERED_ON_REFRESHED_QUOTE	Initial Value = "TO"
AlertTypes.CBOE_TRADE_THROUGH	Initial Value = 1
AlertTypes.NBBO_TRADE_THOUGH	Initial Value = 2
AlertTypes.NO_NBBO_AGENT	Initial Value = 4
AlertTypes.NON_EXECUTION	Initial Value = 3
AlertTypes.SATISFACTION_ALERT	Initial Value = 5
AuctionStates.ABORTED	Initial Value = 4 The auction has been aborted. - NOT currently supported.
AuctionStates.ENDED	Initial Value = 2 The auction has ended. NOT currently supported.

AuctionStates.PREMATURELY_ENDED	Initial Value = 3 The auction has ended prematurely. - NOT currently supported.
AuctionStates.STARTED	Initial Value = 1 The auction has started.
AuctionTypes.AUCTION_HAL	Initial Value = 4 Hybrid Automated Liaison auction type.
AuctionTypes.AUCTION_INTERNALIZATION	Initial Value = 1 Used for auctioning internalization orders.
AuctionTypes.AUCTION_REGULAR_SINGLE	Initial Value = 3 Used for auctioning normal orders (single-leg orders that are not internalized) - NOT currently supported.
AuctionTypes.AUCTION_SAL	Initial Value = 5 Simple Auction Liaison (SAL) is a mechanism that provides a price-improvement auction for simple (non-complex) orders. - NOT CURRENTLY SUPPORTED.
AuctionTypes.AUCTION_STRATEGY	Initial Value = 2 Used for strategy auctions.
AuctionTypes.AUCTION_UNSPECIFIED	Initial Value = 0 NOT currently supported.
AuctionTypes.STOCK_NBBO_FLASH	Initial Value = 6 This attribute is used in the W_STOCK session to indicate (Flash) the Stock NBBO. This is not an auction but it is being used as an AuctionType in order to share the auction event channel. If the user tries to respond with an auction, a reject message will be received.
AuctionTypes.STOCK_ODD_LOT	Initial Value = 7 This new message type is used to indicate the side and size of the odd lot order, as well as the NBBO price to all interested parties.
BillingTypeIndicators.CROSS	Initial Value = C Refers to a trade whereby both buyer and seller are represented on a single transaction. Thus, neither is really a maker or taker per se, but rather virtualy meet one another.
BillingTypeIndicators.CROSS_PRICE_IMP	Initial Value = S Related to trades specifically between the primary and matched orders of an AIM order.

BillingTypeIndicators.FLASH	Initial Value = F Refers to an order that is being presented to the dealers for a short-term auction for step-up, before the order is routed to an away exchange for a fill.
BillingTypeIndicators.FLASH_PRICE_IMP	Initial Value = T Refers to trades between an auctioned order and the incoming response after the autcion started.
BillingTypeIndicators.FLASH_RESPONSE	Initial Value = E Refers to the dealter responding to a flash and effectively stepping up to improve the CBSX market to the revailing price and fulfilling the customer here.
BillingTypeIndicators.FLASH_RESPONSE_PRICE_IMP	Initial Value = U Relates to trades between an auctioned order and the incoming response after the auction started.
BillingTypeIndicators.LINKED_AWAY	Initial Value = X Refers to an order that was sent to another market for execution.
BillingTypeIndicators.LINKED_AWAY_RESPONSE	Initial Value = L Refers to the response from the other exchange filling the CBSX order sent to them.
BillingTypeIndicators.MAKER	Initial Value = A Refers to the person adding liquidity to the market, by basically having an order or quote resting in the book to be traded against.
BillingTypeIndicators.MAKER_TURNER	Initial Value = V Relates to trades between a non-auctioned order and the resting quote or order.
BillingTypeIndicators.ODD_LOT_FLASH	Initial Value = N CBSX odd lot order or the odd lot portion of a mixed lot order that is being flashed.
BillingTypeIndicators.ODD_LOT_RESPONSE	Initial Value = B Response for odd lot orders that are being flashed.
BillingTypeIndicators.OPENING	Initial Value = O Refers to all executions that take place as part of the opening rotation process itself.
BillingTypeIndicators.RESTING	Initial Value = Q Relates to autioned orders and resting quote or orders.
BillingTypeIndicators.RESTING_TURNER	Initial Value = W Relates to the auctioned order and the resting quote or orders.
BillingTypeIndicators.TAKER	Initial Value = R Refers to the person taking liquidity from the market, by basically sending an order to trade against the book. (i.e. they are coming in and taking out the best price)

BookDepthTypes.DELETE_PRICE	Initial Value = 'D'
BookDepthTypes.INSERT_PRICE	Initial Value = 'I'
BookDepthTypes.UPDATE_PRICE	Initial Value = 'U'
ClassStates.CLOSED	Initial Value = 10
ClassStates.ENDING_HOLD	Initial Value = 9
ClassStates.FAST_MARKET	Initial Value = 6
ClassStates.HALTED	Initial Value = 5
ClassStates.NO_SESSION	Initial Value = 7
ClassStates.NOT_IMPLEMENTED	Initial Value = 1
ClassStates.ON_HOLD	Initial Value = 8
ClassStates.OPEN	Initial Value = 4
ClassStates.OPENING_ROTATION	Initial Value = 3
ClassStates.PRE_OPEN	Initial Value = 2
ClassStates.SUSPENDED	Initial Value = 11 Used in a class state change callback event to indicate that a failover situation has occurred affecting the class for which the callback is subscribed and that the class is suspended from trading.
ContingencyTypes.AON	Initial Value = 2 All or None
ContingencyTypes.AUCTION_RESPONSE	Initial Value = 14 Auction response for the order.

ContingencyTypes.AUTOLINK_CROSS	Initial Value = 20 Used for Stock: 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. The matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled.
ContingencyTypes.AUTOLINK_CROSS_MATCH	Initial Value = 21 Used for Stock: 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. The matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled.
ContingencyTypes.BID_PEG_CROSS	Initial Value = 29 Supported in the W_STOCK session. Order price field will be set as "Market." Used to specify on inbound paired orders whether the paried orders should cross at the NBBO bid price or 'x' better than the NBBO bid where 'x' can be designated as a value in pennies including up to 4 decimal places to indicate sub-penny values. The contingency information on the paired orders should have: (1) indication that the paired order set should cross at the NBBO bid (BID_PEG_CROSS) (2) a numeric value stored in the contingency price field representing how many pennies (including up to 4 decimal places to indicate sub-penny values) better than the NBBO bid at which this paired order set should trade. If the specified increment should cause the price to go outside the NBBO (worse than the NBBO), the system should only increment to the other side of the NBBO.

ContingencyTypes.CASH_CROSS	Initial Value = 32 Cash settlement cross order. Supported in the W_STOCK session. Order price field will be set as "Limit."
ContingencyTypes.CLOSE	Initial Value = 12 On close
ContingencyTypes.CROSS	Initial Value = 18 Used for Stock: 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. The matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled.
ContingencyTypes.CROSS_WITHIN	Initial Value = 22 Used for Stock: CURRENTLY UNSUPPORTED Similar to cross orders except that the CROSS_WITHIN order shall trade at or better than the NBBO and within CBOE's market.
ContingencyTypes.DO_NOT_ROUTE	Initial Value = 26 Orders marked DO_NOT_ROUTE will trade immediately or book as long as they do not violate the NBBO. If a DO_NOT_ROUTE order cannot be booked or traded without violating the NBBO, it will be cancelled. This contingency type will be available for any origin type, including "I" orders
ContingencyTypes.FOK	Initial Value = 3 Fill or Kill

ContingencyTypes.INTERMARKET_SWEEP	Initial Value = 15 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.
ContingencyTypes.INTERMARKET_SWEEP_BOOK	Initial Value = 28 Unlike Intermarket_Sweep orders, which are treated like IOC orders, orders marked Intermarket_Sweep_Book will be booked by the system if they cannot be traded.
ContingencyTypes.IOC	Initial Value = 4 Immediate or Cancel
ContingencyTypes.MIDPOINT_CROSS	Initial Value = 17 Used for Stock: 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.. The matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled.
ContingencyTypes.MIN	Initial Value = 6 Minimum - refer to Quantity in contingency field
ContingencyTypes.MIT	Initial Value = 9 Market if touched
ContingencyTypes.NBBO_FLASH_RESPONSE	Initial Value = 27 When a marketable order is received that cannot trade against the CBSX quote due to NBBO restrictions, the order will be flashed at the NBBO. When the auction message appears, users responding to the flash will respond with the new flash response contingency.
ContingencyTypes.NBBO_FLASH_THEN_CANCEL	Initial Value = 25 Orders marked NBBO_FLASH_THEN_CANCEL will be flashed through the flash process if they cannot be executed immediatley at the NBBO. Any part of the order that is not filled will be cancelled after the flash.

ContingencyTypes.NEXT_DAY_CROSS	Initial Value = 33 Next day settlement cross order. Supported in the W_STOCK session. Order price field will be set as "Limit."
ContingencyTypes.NONE	Initial Value = 1 No contingency
ContingencyTypes.NOTHELD	Initial Value = 7 Not held
ContingencyTypes.OFFER_PEG_CROSS	Initial Value = 30 Supported in the W_STOCK session. Order price field will be set as "Market." Used to specify on inbound paired orders whether the paried orders should cross at the NBBO offer price or 'x' better than the NBBO offer price where 'x' can be designated as a value in pennies including up to 4 decimal places to indicate sub-penny values. The contingency information on the paired orders should have: (1) indication that the paired order set should cross at the NBBO offer (OFFER_PEG_CROSS). (2) a numeric value stored in the contingency price field representing how many pennies (including up to 4 decimal places to indicate sub-penny values) better than the NBBO offer at which this paired order set should trade. If the specified increment should cause the price to go outside the NBBO (worse than the NBBO), the system should only increment to the other side of the NBBO.
ContingencyTypes.OPG	Initial Value = 5 Opening
ContingencyTypes.RESERVE	Initial Value = 16 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.
ContingencyTypes.STOCK_ODD_LOT_NBBO_ONLY	Initial Value = 24 Odd lot order for Stock that should be cancelled if it cannot trade at the NBBO.
ContingencyTypes.STP	Initial Value = 10 Stop order. CBSX does not support Stop orders.
ContingencyTypes.STP_LIMIT	Initial Value = 13 Stop limit. CBXS does not support Stop limit orders.

ContingencyTypes.STP_LOSS	Initial Value = 11 Stop loss. CBSX does not support Stop loss orders.
ContingencyTypes.TIED_CROSS	Initial Value = 19 CURRENTLY UNSUPPORTED Used for Stock: Tied Cross: Similar to Cross orders except that they can trade at or better than CBOE's current market and NBBO trade through is allowed. The matching orders (if not arriving as a pair) have to arrive within a specified time otherwise the original cross order will be cancelled.
ContingencyTypes.TIED_CROSS_SWEEP	Initial Value = 31 Supported in the W_STOCK session.
ContingencyTypes.TIED_CROSS_WITHIN	Initial Value = 23 Used for Stock: CURRENTLY UNSUPPORTED Similar to TIED_CROSS orders except that the TIED_CROSS_WITHIN order shall trade at or better than the NBBO and within CBOE's market.
ContingencyTypes.TWO_DAY_CROSS	Initial Value = 34 Two day settlement cross order. Supported in the W_STOCK session. Order price field will be set as "Limit."
ContingencyTypes.WTP	Initial Value = 35 Wash Trade Prevention: When marked with this contingency, the order will be cancelled if it is about to trade against another order or quote from the same user. Additionally, the resting quote or order will be cancelled.
ContingencyTypes.WD	Initial Value = 8 With discretion
CoverageTypes.COVERED	Initial Value = 'C'
CoverageTypes.UNCOVERED	Initial Value = 'U'
CoverageTypes.UNSPECIFIED	Initial Value = 'B'
CurrentMarketTypes.BEST_MARKETS	Initial Value = 1

CurrentMarketTypes.BEST_PUBLIC_MARKETS	Initial Value = 2
CurrentMarketViewtypes.CONTINGENT	Initial Value = 1
CurrentMarketViewtypes.NON_CONTINGENT	Initial Value = 2
CurrentMarketViewtypes.NON_Q	Initial Value = 3
EntryTypes.ADD	Initial Value = 'A'
EntryTypes.BOOK	Initial Value = 'B'
EntryTypes.BUST	Initial Value = 'K'
EntryTypes.CANCEL	Initial Value = 'C'
EntryTypes.CANCEL_REPLACE	Initial Value = 'R'
EntryTypes.FILL	Initial Value = 'F'
EntryTypes.PRICE_ADJUST	Initial Value = 'P'
EntryTypes.STATE	Initial Value = 'S'
EntryTypes.UPDATE	Initial Value = 'U'
ExchangeIndicatorTypes.CLEAR	Initial Value = 21 External exchange's products are available for trading.
ExchangeIndicatorTypes.FAST_MARKET	Initial Value = 23 External exchanges products are in a fast market.
ExchangeIndicatorTypes.HALTED	Initial Value = 22 External exchange's products are halted.
ExchangeIndicatorTypes.OPENING_ROTATION	Initial Value = 24 External exchanges products are in opening rotation

ExchangeMarketInfoType.BBO_ORDER_EXECUTED	Initial Value = 4
ExchangeMarketInfoType.BBO_ORDER_RECEIVED	Initial Value = 3
ExchangeMarketInfoType.NBBO_ORDER_EXECUTED	Initial Value = 2
ExchangeMarketInfoType.NBBO_ORDER_RECEIVED	Initial Value = 1
ExchangeMarketInfoType.WORST_BBO_IN_TIME_WINDOW	Initial Value = 6
ExchangeMarketInfoType.WORST_NBBO_IN_TIME_WINDOW	Initial Value = 5
ExchangeStrings.AMEX	Initial Value = "AMEX" American Stock Exchange
ExchangeStrings.BATS	Initial Value = "BATS" Better Alternative Trading System
ExchangeStrings.BSE	Initial Value = "BSE" Boston Stock Exchange
ExchangeStrings.BYX	Initial Value = "BYX" National Better Alternative Trading
ExchangeStrings.CBOE	Initial Value = "CBOE" Chicago Board Options Exchange
ExchangeStrings.CBOE2	Initial Value = "CBOE2" Chicago Board Options Exchange 2
ExchangeStrings.CBOT	Initial Value = "CBOT" Chicago Board of Trade
ExchangeStrings.CFE	Initial Value = "CFE" CBOE Futures Exchange
ExchangeStrings.CHX	Initial Value = "CHX" Chicago Stock Exchange
ExchangeStrings.CME	Initial Value = "CME" Chicago Mercantile Exchange.
ExchangeStrings.CSE	Initial Value = "CSE" Cincinnati Stock Exchange.

ExchangeStrings.EDGA	Initial Value = "EDGA" Non-ADF Exchange
ExchangeStrings.EDGX	Initial Value = "EDGX" Non-ADF Exchange
ExchangeStrings.ISE	Initial Value = "ISE" International Stock Market
ExchangeStrings.LIFFE	Initial Value = "LIFFE" International Financial Futures and Options Exchange
ExchangeStrings.NASD	Initial Value = "NASD" National Association of Security Dealers
ExchangeStrings.NASDAQ	Initial Value = "NASDAQ" National Association of Securities Dealers Automated
ExchangeStrings.NQLX	Initial Value = "NQLX" Nasdaq Liffe Markets
ExchangeStrings.NSX	Initial Value = "NSX" National Stock Exchange
ExchangeStrings.NYME	Initial Value = "NYME" New York Mercantile Exchange
ExchangeStrings.NYSE	Initial Value = "NYSE" New York Stock Exchange
ExchangeStrings.ONE	Initial Value = "ONE" OneChicago Exchange
ExchangeStrings.PHLX	Initial Value = "PHLX" Philadelphia Stock Exchange
ExchangeStrings.PSE	Initial Value = "PSE" Pacific Stock Exchange
ExpectedOpeningPriceTypes.DPM_QUOTE_INVALID	Initial Value = 9 Invalid DPM quote is entered.
ExpectedOpeningPriceTypes.MORE_BUYERS	Initial Value = 2 This type has been deprecated.
ExpectedOpeningPriceTypes.MORE_SELLERS	Initial Value = 3 This type has been deprecated.
ExpectedOpeningPriceTypes.MULTIPLE_OPENING_PRICES	Initial Value = 5

ExpectedOpeningPriceTypes.NEED_DPM_QUOTE_TO_OPEN	Initial Value = 8
ExpectedOpeningPriceTypes.NEED_MORE_BUYERS	Initial Value = 3 Indicates that the current market is imbalanced and will need more buyers' contracts (n) at the opening price. The message will provide users with the potential opening price and the imbalanced quantity (n).
ExpectedOpeningPriceTypes.NEED_MORE_SELLERS	Initial Value = 2 Indicates that the current market is imbalanced and will need more sellers' contracts (m) at the opening price. The message will provide users with the potential opening price and the imbalanced quantity (m).
ExpectedOpeningPriceTypes.NEED_QUOTE_TO_OPEN	Initial Value = 6
ExpectedOpeningPriceTypes.NO_OPENING_TRADE	Initial Value = 4 Message will only be sent if the previous message was something other than value 4.
ExpectedOpeningPriceTypes.OPENING_PRICE	Initial Value = 1
ExpectedOpeningPriceTypes.PRICE_NOT_IN_QUOTE_RANGE	Initial Value = 7
ExpectedOpeningPriceTypes.PRICE_NOTE_IN_BOTR_RANGE	Initial Value = 10 Indicates that the potential opening price is not in the valid Best of the Rest (BOTR) range based on the current BOTR price, current potential opening quantity and the system configuration. The message will provide users with the potential opening price and opening quantity.
ExpirationStyles.AMERICAN	Initial Value = 'A'
ExpirationStyles.EUROPEAN	Initial Value = 'E'
ExtensionFields.ADJUSTED_PRICE_INDICATOR	Initial Value = "5205" Used for Linkage.
ExtensionFields.ASSOCIATED_ORDER_ID	Initial Value = "assocOrderId" Customer orders associated with Linkage orders.
ExtensionFields.AUCTION_ID	Initial Value = "auctionId" Used for auction response.

ExtensionFields.AUTO_EXECUTION_SIZE	Initial Value = "5201" Used for Linkage.
ExtensionFields.AWAY_CANCEL_REPORT_EXEC_ID	Initial Value = "awayCancelReportExecId"
ExtensionFields.AWAY_EXCHANGE_EXEC_ID	Initial Value = "17" Used for Linkage.
ExtensionFields.AWAY_EXCHANGE_ORDER_ID	Initial Value = "37" Used for Linkage.
ExtensionFields.AWAY_EXCHANGE_TRANSACT_TIME	Initial Value = "60" Used for Linkage.
ExtensionFields.AWAY_EXCHANGE_USER_ACRONYM	Initial Value = "1" Used for Linkage.
ExtensionFields.BARTID	Initial Value = "BARTID" Used for routing an order to a BART terminal.
ExtensionFields.BILLING_TYPE	Initial Value = "billingType" Used for CBSX billing
ExtensionFields.CBOE_EXEC_ID	Initial Value = "cboeExecId" Used for Linkage.
ExtensionFields.CMTA_FIRM	Initial Value = "cmta" Used for COB auction messages.
ExtensionFields.CORRESPONDENT_FIRM	Initial Value = "corresfirm" Used for COB auction messages.
ExtensionFields.DIRECTED_FIRM	Initial Value = "dfirm" Used for Directed AIM auctions.
ExtensionFields.DPM	Initial Value = "DDPM" Used for Directed AIM auctions.
ExtensionFields.EXCHANGE_DESTINATION	Initial Value = "100" Used for Linkage.
ExtensionFields.EXEC_BROKER	Initial Value = "execBroker"
ExtensionFields.EXECUTING_FIRM	Initial Value = "firm" Used for COB auction messages.
ExtensionFields.EXECUTION_RECEIPT_TIME	Initial Value = "5207" Used for Linkage.

ExtensionFields.EXPIRATION_TIME	Initial Value = "ExpirationTime"
ExtensionFields.FADE_EXCHANGE	Initial Value = "FADE_EXCHANGE"
ExtensionFields.GUICFI	Initial Value = "guicfi" Used for COB auction messages.
ExtensionFields.HANDLING_INSTRUCTION	Initial Value = "21" Used for Linkage.
ExtensionFields.LINKAGE_MECHANISM	Initial Value = "LinkageMechanism" Used for Linkage.
ExtensionFields.MEET_FIRM_NAME	Initial Value = "9381" Firm information for stock leg of a buy-write
ExtensionFields.MEET_LOCATION_IN	Initial Value = "9380" Used for stock leg of a buy-write
ExtensionFields.MEET_LOCATION_OUT	Initial Value = "207" Used for stock leg of a buy-write
ExtensionFields.NBBO_ASK_PRICE	Initial Value = "nbboask" Used for COB auction messages
ExtensionFields.NBBO_BID_PRICE	Initial Value = "nbbobid" Used for COB auction messages.
ExtensionFields.OLA_REJECT_REASON	Initial Value = "5209" Used for Linkage.
ExtensionFields.ORDER_CAPACITY	Initial Value = "6528" Used for Linkage.
ExtensionFields.ORDER_RESTRICTIONS	Initial Value = "6529" Used for Linkage.
ExtensionFields.ORIGINAL_ORDER_TIME	Initial Value = "5208" Used for Linkage.
ExtensionFields.ORIGINAL_QUANTITY	Initial Value = "originalQuantity"
ExtensionFields.ORS_ID	Initial Value = "orsId"
ExtensionFields.PDPM	Initial Value = "DPMM" Used for Directed AIM auctions.

ExtensionFields.SATISFACTION_ALERT_ID	Initial Value = "satAlertId" Associates an outbound S order with the alert.
ExtensionFields.SATISFACTION_ORDER_DISPOSITION	Initial Value = "5206" Used for Linkage.
ExtensionFields.SIDE	Initial Value = "side"
ExtensionFields.STOCK_FIRM	Initial Value = "STOCK_FIRM" Firm information for stock leg of a buy-write.
ExtensionFields.STOCK_FIRM_NAME	Initial Value = "STOCK_FIRM_NAME" Firm information for stock leg of a buy-write.
ExtensionFields.TEXT	Initial Value = "58" Used for Linkage.
ExtensionFields.TRADE_THRU_PRICE	Initial Value = "5204" Used for Linkage.
ExtensionFields.TRADE_THRU_SIZE	Initial Value = "5203" Used for Linkage.
ExtensionFields.TRADE_THRU_TIME	Initial Value = "5202" Used for Linkage.
ExtensionFields.USER_ASSIGNED_CANCEL_ID	Initial Value = "11" Used for Linkage.
FillRejectReasons.INVALID_PRICE	Initial Value = ActivityReasons::INVALID_PRICE
FillRejectReasons.INVALID_PRODUCT	Initial Value = ActivityReasons::INVALID_PRODUCT
FillRejectReasons.INVALID_QUANTITY	Initial Value = ActivityReasons::INVALID_QUANTITY
FillRejectReasons.INVALID_SIDE	Initial Value = ActivityReasons::INVALID_SIDE
FillRejectReasons.NO_MATCHING_ORDER	Initial Value = ActivityReasons::NO_MATCHING_ORDER
FillRejectReasons.OTHER	Initial Value = ActivityReasons::OTHER
FillRejectReasons.STALE_EXECUTION	Initial Value = ActivityReasons::STALE_EXECUTION

HandlingInstructions.AUTOMATIC	Initial Value = 1
HandlingInstructions.MANUAL	Initial Value = 2
HandlingInstructions.OTHER	Initial Value = 3
LinkageMechanisms.NON_CBOE_LINKAGE_MECHANISM	Initial Value = 1
ListingStates.ACTIVE	Initial Value = 1 This signifies that the product is listed for trading.
ListingStates.INACTIVE	Initial Value = 2 Initial value=2. This signifies that the product is NOT listed for trading. CBOE will maintain inactive productes for a configured retention period (usually 5 days), and then move them into the obsolete state.
ListingStates.OBSOLETE	Initial Value = 4 Initial value=4. This is a temporary server state as a product goes from active/inactive to unlisted. It should not be seen by CMi users normally. Obsolete indicates that the product is ready to be removed from the CBOEdirect system on the next cleanup.
ListingStates.UNLISTED	Initial Value = 3 Initial value=3. This is a listing state for a product that will not be available to download and is not eligible for trading. API users will not see a product with this listing state. When CBOE creates a new product but it has not yet been listed for trading, CBOE assignes it to the unlisted state. A product can reside in the CBOEdirect system indefinitely with the unlisted state. In this sense, unlisted is the same as inactive.
LoginSessionModes.NETWORK_TEST	Initial Value = 2 This login session mode indicates that the client is expecting to communicate with a real CAS that is running in a test mode. An exception will be thrown if the CAS is not in a test mode.
LoginSessionModes.PRODUCTION	Initial Value = 3 This login session mode is used to connect to a production CAS. An exception will be thrown if the CAS is not in production mode.

LoginSessionModes.STAND_ALONE_TEST	Initial Value = 1 Stand alone testing mode. In this mode the client application communicates with a CAS simulator. An exception will be thrown if an attempt is made to connect to real CAS.
LoginSessionTypes.PRIMARY	Initial Value = 1 Login being requested will be the primary login for the client.
LoginSessionTypes.SECONDARY	Initial Value = 2 Login requested is a secondary - for fail over.
MarketChangeReasons.COMBINED	Initial Value = 3
MarketChangeReasons.EXCHANGE	Initial Value = 1
MarketChangeReasons.NBBO	Initial Value = 2
MarketDataHistoryEntryTypes.EXPECTED_OPEN_PRICE	Initial Value = 3
MarketDataHistoryEntryTypes.MARKET_CONDITION_ENTRY	Initial Value = 4
MarketDataHistoryEntryTypes.PRICE_REPORT_ENTRY	Initial Value = 2
MarketDataHistoryEntryTypes.QUOTE_ENTRY	Initial Value = 1
MarketDataHistoryEntryTypes.UNSIZED_QUOTE_ENTRY	Initial Value = 5
MarketIndicators.ADDITIONAL_INFO_REL_SEC	Initial Value = 27
MarketIndicators.ADDITIONAL_INFORMATION	Initial Value = 23
MarketIndicators.ASK_IS_BOOK	Initial Value = 4
MarketIndicators.AUTO_EXECUTION	Initial Value = 2
MarketIndicators.BID_ASK_IS_BOOK	Initial Value = 5

MarketIndicators.BID_IS_BOOK	Initial Value = 3
MarketIndicators.CLOSED_MARKET_MAKER	Initial Value = 22
MarketIndicators.CLOSING	Initial Value = 14
MarketIndicators.DEPTH_BID_OFFER	Initial Value = 18
MarketIndicators.DEPTH_ON_BID	Initial Value = 13
MarketIndicators.DEPTH_ON_OFFER	Initial Value = 12
MarketIndicators.DISQUALIFIED	Initial Value = 10
MarketIndicators.EQUIPMENT_CHANGOVER	Initial Value = 31
MarketIndicators.FAST_MARKET	Initial Value = 8
MarketIndicators.HALT_IN_VIEW_COMMON	Initial Value = 30
MarketIndicators.HALT_REL_SEC_NEWS_DISS	Initial Value = 20
MarketIndicators.HALT_REL_SEC_NEWS_PEND	Initial Value = 21
MarketIndicators.INACTIVE	Initial Value = 6
MarketIndicators.NEWS_DISSEMINATION	Initial Value = 15
MarketIndicators.NEWS_PENDING	Initial Value = 26
MarketIndicators.NO_OPEN_RESUME	Initial Value = 32
MarketIndicators.NON_FIRM_QUOTE	Initial Value = 24

MarketIndicators.OPENING	Initial Value = 25
MarketIndicators.ORDER_IMBALANCE	Initial Value = 19
MarketIndicators.ORDER_INFLUX	Initial Value = 16
MarketIndicators.PRE_OPENING_INDICATION	Initial Value = 17
MarketIndicators.REGULAR	Initial Value = 1
MarketIndicators.ROTATION	Initial Value = 7
MarketIndicators.SLOW_ON_ASK_SIDE	Initial Value = 34
MarketIndicators.SLOW_ON_BID_SIDE	Initial Value = 33
MarketIndicators.SLOW_ON_BOTH_SIDES	Initial Value = 35
MarketIndicators.TRADING_HALT	Initial Value = 9
MarketIndicators.UNKNOWN	Initial Value = 11
MatchTypes.AUTO_MATCH	Initial Value = 3 The internalization order can be set to AUTO_MATCH by using the value 3.
MatchTypes.GUARANTEE_STARTING_PRICE	Initial Value = 1 This match type is not currently supported.
MatchTypes.LIMIT_PRICE	Initial Value = 2 Used to enter a limit price for the internalization order.
MatchTypes.UNSPECIFIED	Initial Value = 0 This constant is used as the default match type for internalization. - NOT currently supported.
MessageKeyTypes.ITS_GENERAL_ADMIN	Initial Value = 1

MessageKeyTypes.ITS_PRE_OPEN_CANCEL	Initial Value = 5
MessageKeyTypes.ITS_PRE_OPEN_INDICATION	Initial Value = 2
MessageKeyTypes.ITS_PRE_OPEN_RESPONSE	Initial Value = 3
MessageKeyTypes.ITS_PRE_OPEN_SECOND_LOOK	Initial Value = 4
MultiplePartiesIndicators.NO	Initial Value = 2
MultiplePartiesIndicators.UNKNOWN	Initial Value = 3
MultiplePartiesIndicators.YES	Initial Value = 1
OptionalDataFields.NAMES_LATER	Initial Value = NLTR Used for Stock: Traders can indicate a non real-time clearing ("Names Later") order using const OptionalDataField NAMES_LATER = "NLTR". If the order gets executed, the trade will not automatically clear. The trader will have to provide the contra-party information at the end of the day before the trade clears.
OptionTypes.CALL	Initial Value = 'C'
OptionTypes.PUT	Initial Value = 'P'
OrderBookPriceViewTypes.BY_ORIGIN_TYPE	Initial Value = 1 Order book price by origin type.
OrderBookStatusValues.ORDER_BOOK_OPENING_ROTATION_LOCKED	Initial Value = 1
OrderBookStatusValues.ORDER_BOOK_OPENING_ROTATION_UNLOCKED	Initial Value = 2
OrderBookStructTradableTypes.BOOK_ITEM_ORDER	Initial Value = 'O'
OrderBookStructTradableTypes.BOOK_ITEM_QUOTE	Initial Value = 'Q' Quote side

OrderBookStructTradableTypes.BOOK_ITEM_QUOTE_TRIGGER	Initial Value = 'T'
OrderCancelTypes.CANCEL_ALL_QUANTITY	Initial Value = 3 This cancellation type cancels all the remaining open quantity of an order. Available to support FIX style order cancel requests.
OrderCancelTypes.DESIRED_CANCEL_QUANTITY	Initial Value = 1 This type of cancellation uses the quantity in the cancel request as the quantity of the order to be cancelled.
OrderCancelTypes.DESIRED_REMAINING_QUANTITY	Initial Value = 2 This type of cancellation uses the quantity as the desired remaining quantity of the order after cancellation. It is not the quantity to be cancelled. This type of cancellation is compatible with FIX order handling semantics.
OrderFlowDirectionType.BOTH	Initial Value = 3 The order is both inbound and outbound.
OrderFlowDirectionType.INBOUND	Initial Value = 1 The order is inbound.
OrderFlowDirectionType.OUTBOUND	Initial Value = 2 The order is outbound.
OrderNBBOProtectionTypes.FULL	Initial Value = 2
OrderNBBOProtectionTypes.NONE	Initial Value = 1
OrderOrigins.BROKER_DEALER	Initial Value = 'B'
OrderOrigins.BROKER_DEALER_FBW_ICM	Initial Value = 'K' For CBSX, K orders will not link away. Any remaining balance that should link away will be cancelled.
OrderOrigins.MANUAL_QUOTE_ORDER	Initial Value = 'K' PAR client users submit K orders as manual quote orders.
OrderOrigins.BROKER_DEALER_FBW_NON_CUSTOMER	Initial Value = 'W'
OrderOrigins.CTI1Origin1	Initial Value = 'V' Member, customer segregated account.

OrderOrigins.CTI1Origin2	Initial Value = 'E' Member, house account.
OrderOrigins.CTI1Origin5	Initial Value = 'Q' Member, SIPC protected account.
OrderOrigins.CTI3Origin1	Initial Value = 'G' User proxy for trader, customer segregated account.
OrderOrigins.CTI3Origin2	Initial Value = 'H' User proxy for trader, house account.
OrderOrigins.CTI3Origin5	Initial Value = 'R' User proxy for trader, SIPC protected account.
OrderOrigins.CTI4Origin2	Initial Value = 'O' Non-member, house account.
OrderOrigins.CTI4Origin5	Initial Value = 'T' Non-member SIPC protected account.
OrderOrigins.CUSTOMER	Initial Value = 'C' CTI Equivalent - non-member, customer segregated account.
OrderOrigins.CUSTOMER_BROKER_DEALER	Initial Value = 'X'
OrderOrigins.CUSTOMER_FBW	Initial Value = 'D'
OrderOrigins.FIRM	Initial Value = 'F' CTI Equivalent - firm trader, house account.
OrderOrigins.FIRM_FBW_ICM	Initial Value = 'J' For CBSX, J orders will not link away. Any remaining balance that should link away will be cancelled.
OrderOrigins.FIRM_FBW_NON_CUSTOMER	Initial Value = 'L'
OrderOrigins.ITS_POR	Initial Value = 'a' Order on behalf of ITS pre-opening response from away exchange.
OrderOrigins.M_N_Y_FBW	Initial Value = 'U' Market-Maker using a floor broker workstation.
OrderOrigins.MARKET_MAKER	Initial Value = 'M'

OrderOrigins.MARKET_MAKER_AWAY	Initial Value = 'N'
OrderOrigins.MARKET_MAKER_IN_CROWD	Initial Value = 'I' The "I" order origin type is an in crowd Market Maker order that is treated like a quote in the book. Unlike quotes, "I" orders are not cancelled at system logout.
OrderOrigins.N_Y_FBW	Initial Value = 'Z' Non-CBOE Member or Specialist Away.
OrderOrigins.PRINCIPAL	Initial Value = 'P' A market maker order origin type for Linkage.
OrderOrigins.PRINCIPAL_ACTING_AS_AGENT	Initial Value = 'A' A market maker order origin type for linkage.
OrderOrigins.SATISFACTION	Initial Value = 'S' A market maker order origin type for Linkage.
OrderOrigins.UNDERLY_SPECIALIST	Initial Value = 'Y'
OrderStates.ACTIVE	Initial Value = 6 CBOE currently does not use this designation.
OrderStates.BOOKED	Initial Value = 1 The order was entered into the CBOE book.
OrderStates.CANCEL	Initial Value = 2 An order will have an order state of CANCEL only if the last action on the order is a cancellation. If an order is partially canceled and partially filled, but a portion of the order still remains in the book, the order state will be BOOKED. If the remaining BOOKED quantity is filled, then the order state will be FILL. If the remaining BOOKED quantity is canceled, then the order state will be CANCEL. If any portion of an order remains BOOKED at the close of trading, CBOE will automatically cancel the balance of that order, and change the order state to CANCEL. At the end of the day processing (around 4:00 p.m. Central Time), CBOE changes the state of the order from CANCEL to PURGED. PURGED orders will remain in the CBOE system until the end of the following business day.
OrderStates.EXPIRED	Initial Value = 7 CBOE currently does not use this designation.

OrderStates.FILL	Initial Value = 3 An order will have an order state of FILL only if the last action on the order is a fill report. If an order is partially canceled and partially filled, but a portion of the order still remains in the book, the order state will be BOOKED. If the remaining BOOKED quantity is filled, then the order state will be FILL. If the remaining BOOKED quantity on a partially filled order is canceled, then the order state will be CANCEL.
OrderStates.INACTIVE	Initial Value = 5 CBOE currently does not use this designation.
OrderStates.OPEN_OUTCRY	Initial Value = 4 CBOE currently does not use this designation.
OrderStates.PURGED	Initial Value = 8 During the trading day, or at the close of trading, an order moves into the CANCEL state. At the end of the day processing (around 4:00 p.m. Central time), CBOE changes the state of the order from CANCEL to PURGED. PURGED orders will remain in the CBOE system until the end of the following business day.
OrderStates.REMOVED	Initial Value = 9 CBOE currently does not use this designation.
OrderStates.WAITING	Initial Value = 10 For purely electronic trading sessions such as ONE_MAIN and CFE_MAIN, the order state WAITING is the state that CBOE uses when a STP or STP Limit order is accepted by CBOE. When the order stop condition is met, the STP or STP Limit order will be triggered and it will either be immediately traded or it will be entered into the CBOE book. If it is entered into the book, CBOE will publish an order status update with the order state set to BOOKED. When the STP or STP Limit order is triggered, if it is partially filled, CBOE will publish a fill report, but the order will still have an order state of BOOKED. When the STP or STP Limit order is triggered, if it is entirely filled, CBOE will publish a fill report, and the order will have an order state of FILL. STP and STP Limit orders in Hybrid and non-Hybrid classes in the W_MAIN session will never receive an order status of WAITING. When a firm enters a STP or STP Limit order, CBOE will report the order with a status of BOOKED, regardless of whether the order is triggered or not.
OverrideIndicatorTypes.BOOK_OVERRIDE	Initial Value = 'B' Used by TPF. Manually enter last sale in ebook.

OverrideIndicatorTypes.LINKAGE	Initial Value = 'L' Used by TPF. Last Sale displays as a Linkage order on MDR.
OverrideIndicatorTypes.NONE	Initial Value = ' ' Used by TPF. Normal last sale on the MDR display.
OverrideIndicatorTypes.OFFER_OVERRIDE	Initial Value = 'O' Used by TPF. Manually override offer.
OverrideIndicatorTypes.SUPERVISORY_OVERRIDE	Initial Value = 'X' Used by TPF. Manual override of price.
PositionEffects.CLOSED	Initial Value = 'C'
PositionEffects.NOTAPPLICABLE	Initial Value = 'N'
PositionEffects.OPEN	Initial Value = 'O'
PreOpeningIndicationTypes.CANCEL	Initial Value = 2 Cancels pre-opening notification.
PreOpeningIndicationTypes.INDICATION	Initial Value = 1 Notifies ITS participants of the opening.
PreOpeningIndicationTypes.SEC_LOOK	Initial Value = 3 Second look sent after the opening for a better price.
PriceAdjustmentActions.PRICE_ADJUSTMENT_CREATE	Initial Value = 3
PriceAdjustmentActions.PRICE_ADJUSTMENT_DELETE	Initial Value = 2
PriceAdjustmentActions.PRICE_ADJUSTMENT_MOVE	Initial Value = 4
PriceAdjustmentActions.PRICE_ADJUSTMENT_UPDATE	Initial Value = 1
PriceAdjustmentTypes.COMMON_DISTRIBUTION	Initial Value = 8
PriceAdjustmentTypes.DIVIDEND_CASH	Initial Value = 2
PriceAdjustmentTypes.DIVIDEND_PERCENT	Initial Value = 3

PriceAdjustmentTypes.DIVIDEND_STOCK	Initial Value = 4
PriceAdjustmentTypes.LEAP_ROLLOVER	Initial Value = 5
PriceAdjustmentTypes.MERGER	Initial Value = 6
PriceAdjustmentTypes.SPLIT	Initial Value = 1
PriceAdjustmentTypes.SYMBOL_CHANGE	Initial Value = 7
PriceConstants.NO_PRICE	Initial Value = -2147483648
PriceDisplayTypes.DECIMAL	Initial Value = 2
PriceDisplayTypes.FRACTION	Initial Value = 1
PriceScale.DEFAULT_SCALE	Initial Value = 1000000000
PriceTypes.CABINET	Initial Value = 4 Used for cabinet orders
PriceTypes.LIMIT	Initial Value = 2 Used for limit orders and quotes.
PriceTypes.MARKET	Initial Value = 3 Used for market orders. For CBSX, market orders will not be allowed unless the product is in OPEN or FAST state.
PriceTypes.NO_PRICE	Initial Value = 1 Only used to describe something that has no price such as a single sided quote or Market Data callback that has no price.
PriceTypes.VALUED	Initial Value = 2 Used for limit orders and quotes. This is the exact same as LIMIT.
ProductClass.DEFAULT_CLASS_KEY	Initial Value = 0 Default class key value

ProductStates.CLOSED	Initial Value = 1
ProductStates.ENDING_HOLD	Initial Value = 9 Product hold is ending
ProductStates.FAST_MARKET	Initial Value = 6
ProductStates.HALTED	Initial Value = 5
ProductStates.NO_SESSION	Initial Value = 7 Product has not been assigned to a trading session
ProductStates.ON_HOLD	Initial Value = 8 Product has been placed on trading hold
ProductStates.OPEN	Initial Value = 4
ProductStates.OPENING_ROTATION	Initial Value = 3
ProductStates.PRE_OPEN	Initial Value = 2
ProductStates.SUSPENDED	Initial Value = 10 Indicates that a failover situation has occurred affecting the product for which the callback is subscribed and that the product is suspended from trading.
ProductStates.UNKNOWN	Initial Value = 0
ProductTypes.COMMODITY	Initial Value = 1
ProductTypes.DEBT	Initial Value = 2
ProductTypes.EQUITY	Initial Value = 3
ProductTypes.FUTURE	Initial Value = 4
ProductTypes.INDEX	Initial Value = 5

ProductTypes.INTEREST_RATE_COMPOSITE	Initial Value = 12
ProductTypes.LINKED_NOTE	Initial Value = 6
ProductTypes.OPTION	Initial Value = 7
ProductTypes.STRATEGY	Initial Value = 11
ProductTypes.UNIT_INVESTMENT_TRUST	Initial Value = 8
ProductTypes.VOLATILITY_INDEX	Initial Value = 9
ProductTypes.WARRANT	Initial Value = 10
QueryDirections.QUERY_BACKWARD	Initial Value = 2
QueryDirections.QUERY_FORWARD	Initial Value = 1
QueueActions.DISCONNECT_CONSUMER	Initial Value = '3' Automatically unregisters the callback object when the queue size is exceeded. Currently, the queue size value for the Hybrid CAS is configured to 5,000.
QueueActions.FLUSH_QUEUE	Initial Value = '1' Allows the end user to continue building large queues on the CAS. Once the queue reaches a certain size, it will automatically be flushed then allowed to rebuild. The queue size is currently configured to 1,000.
QueueActions.NO_ACTION	Initial Value = '0' CBOE currently does not support this designation. It will return the same resulting actions as DISCONNECT_CONSUMER.
QueueActions.OVERLAY_LAST	Initial Value = '2' The callback object receives the most recent market data information for a product, overlaying updates that are in progress.
QuoteUpdateControlValues.CONTROL_DISABLED	Initial Value = 0 Remove the ability to update a quote.

ReportTypes.CANCEL_ORDER_REQUEST	Initial Value = 103 This report type is not used in the CancelReportStruct.
ReportTypes.CANCEL_ORDER_REQUEST_REJECT	Initial Value = 104
ReportTypes.CANCEL_ORDER_REQUEST_REJECT_REJECTED	Initial Value = 108
ReportTypes.CANCEL_REPORT_REJECT	Initial Value = 105
ReportTypes.CANCEL_REPORT_REJECT_REJECTED	Initial Value = 109
ReportTypes.FILL_REJECT	Initial Value = 102
ReportTypes.FILL_REJECT_REJECTED	Initial Value = 107
ReportTypes.NEW_ORDER_REJECT	Initial Value = 101
ReportTypes.NEW_ORDER_REJECT_REJECTED	Initial Value = 106
ReportTypes.REGULAR_REPORT	Initial Value = 1
ReportTypes.STRATEGY_LEG_REPORT	Initial Value = 3
ReportTypes.STRATEGY_REPORT	Initial Value = 2
RFQTypes.MANUAL	Initial Value = 1
RFQTypes.SYSTEM	Initial Value = 2
SatisfactionOrderDispositions.ORDER_SIZE_MORE_THAN_TRADE_THROUGH_SIZE	Initial Value = 2 Cancel - order size is more than the trade through size.
SatisfactionOrderDispositions.PRO_RATA_DISTRIBUTION	Initial Value = 1 Cancel due to pro-rata distribution.
SatisfactionOrderDispositions.SATISFIED_AS_SPECIFIED	Initial Value = 0 Order was satisfied.

SatisfactionOrderRejectReasons.COMM_DELAYS	Initial Value = ActivityReasons::COMM_DELAYS
SatisfactionOrderRejectReasons.CROWD_TRADE	Initial Value = ActivityReasons::CROWD_TRADE
SatisfactionOrderRejectReasons.INVALID_PRODUCT_TYPE	Initial Value = ActivityReasons::INVALID_PRODUCT_TYPE
SatisfactionOrderRejectReasons.LATE_PRINT	Initial Value = ActivityReasons::LATE_PRINT
SatisfactionOrderRejectReasons.NON_BLOCK_TRADE	Initial Value = ActivityReasons::NON_BLOCK_TRADE
SatisfactionOrderRejectReasons.ORIGINAL_ORDER_REJECTED	Initial Value = ActivityReasons::ORIGINAL_ORDER_REJECTED
SatisfactionOrderRejectReasons.PROCESSING_PROBLEMS	Initial Value = ActivityReasons::PROCESSING_PROBLEMS
SatisfactionOrderRejectReasons.TRADE_BUSTED	Initial Value = ActivityRejectReasons::TRADE_BUSTED
SatisfactionOrderRejectReasons.TRADE_REJECTED	Initial Value = ActivityReasons::TRADE_REJECTED
SessionNameValues.ALL_SESSION_NAME	Initial Value = "ALL_SESSIONS"
Sides.AS_DEFINED	Initial Value = 'D'
Sides.ASK	Initial Value = 'A'
Sides.BID	Initial Value = 'B'
Sides.BUY	Initial Value = 'B'
Sides.BUY_MINUS	Initial Value = 'M'
Sides.OPPOSITE	Initial Value = '0'
Sides.SELL	Initial Value = 'S'

Sides.SELL_PLUS	Initial Value = 'P'
Sides.SELL_SHORT	Initial Value = 'H' Used for Stock.
Sides.SELL_SHORT_EXEMPT	Initial Value = 'X' Used for Stock.
Sides.UNSPECIFIED	Initial Value = 'U' Used when the side of the order is not specified.
Sources.COMPASS	Initial Value = 'C' Order processed by the COMPASS system.
Sources.ITS	Initial Value = 'I' Order was processed by the Intermarket Trading System.
Sources.LINKAGE	Initial Value = 'L' Order processed by the Linkage system.
Sources.SBT	Initial Value = 'S' Order processed by the SBT system.
Sources.TPF	Initial Value = 'T' Order processed by the TPF system.
StatusUpdateReasons.BOOKED	Initial Value = 1 CBOE entered the order or quote into the CBOE book.
StatusUpdateReasons.BUST	Initial Value = 8 CBOE partially or completely busted (or bust-reinstated) an order or quote trade.
StatusUpdateReasons.CANCEL	Initial Value = 2 The user or CBOE has partially or completely cancelled the order or quote.
StatusUpdateReasons.FILL	Initial Value = 3 The order or quote is partially or completely filled.
StatusUpdateReasons.NEW	Initial Value = 7 CBOE created a new order or quote.
StatusUpdateReasons.OPEN_OUTCRY	Initial Value = 6 CBOE currently does note use this designation.

StatusUpdateReasons.POSSIBLE_RESEND	Initial Value = 10 CBOE may or may not be resending the order or quote status message. CBOE requires firms to set the GMD field to "TRUE" in their order and quote status subscriptions to avoid unnecessary resending of unacknowledged messages with the POSSIBLE_RESEND designation.
StatusUpdateReasons.QUERY	Initial Value = 4 When a firm initially subscribes for orders or quotes, CBOE sends the firm a copy of each order or quote. Since the firm is not receiving the order or quote because of a market event (fill, cancel, bust, etc.), CBOE places the state QUERY on each order or quote.
StatusUpdateReasons.QUOTE_TRIGGER_BUY	Initial Value = 11 Quote trigger on the buy side of the quote.
StatusUpdateReasons.QUOTE_TRIGGER_SELL	Initial Value = 12 Quote trigger on the sell side of the quote.
StatusUpdateReasons.REINSTATE	Initial Value = 9 When an order bust-reinstate takes place, CBOE busts all (or a portion) of an order trade, and then reinstates (re-enters) the user's order (or a portion of the user's order) into the book. During an order bust-reinstate, the order trade bust and the order reinstate will always take place at the exact same time. The reinstate quantity will always equal the bust quantity.
StatusUpdateReasons.UPDATE	Initial Value = 5 This is the resultant status when the firm successfully calls the acceptOrderUpdateRequest message.
StrategyTypes.BUY_WRITE	Initial Value = 9
StrategyTypes.COMBO	Initial Value = 8
StrategyTypes.DIAGONAL	Initial Value = 7
StrategyTypes.PSEUDO_STRADDLE	Initial Value = 3
StrategyTypes.RATIO	Initial Value = 5
StrategyTypes.STRADDLE	Initial Value = 2

StrategyTypes.TIME	Initial Value = 6
StrategyTypes.UNKNOWN	Initial Value = 1
StrategyTypes.VERTICAL	Initial Value = 4
TickDirectionTypes.MINUS_TICK	Initial Value = '-' MINUS_TICK '-' Means the current last sale price is lower than the previous last sale price. Also known as a "downtick".
TickDirectionTypes.PLUS_TICK	Initial Value = '+' PLUS_TICK '+' Means the current last sale price is higher than the previous last sale price. Also known as an "uptick".
TickDirectionTypes.UNKNOWN_TICK	Initial Value = ' ' UNKNOWN_TICK ' 'Means the tick direction field is not used or not known. For options, the tick direction field is always set to UNKNOWN_TICK.
TickDirectionTypes.ZERO_MINUS_TICK	Initial Value = '_' ZERO_MINUS_TICK '_' Means the current last sale price is the same as the previous last sale price, but is lower than the most recent different last sale price.
TickDirectionTypes.ZERO_PLUS_TICK	Initial Value = '*' ZERO_PLUS_TICK '*' Means the current last sale price is the same as the previous last sale price, but is higher than the most recent different last sale price.
TimesInForce.ALL	Initial Value = 'A' Stands for all orders (i.e. GTC+DTD+DAY)
TimesInForce.DAY	Initial Value = 'D' Day order
TimesInForce.GTC	Initial Value = 'G' Good Til Cancel orders. CBSX does not support GTC orders.
TimesInForce.GTD	Initial Value = 'T' Good until datetime

TradeReportHandlingInstructions.NO_TICKER	Initial Value = 2 Indicates not to publish to Ticker. No last sales data will go to COPP and CFN. Used by One Chicago for block trade functionality.
TradeReportHandlingInstructions.NO_TRADEREPORT	Initial Value = 1 Indicates not to publish the trade to CTM. No trade reports will go to CTM-R and OCC. Used by OneChicago for block trade functionality.
TradeReportHandlingInstructions.NO_TRADEREPORT_NO_TICKER	Initial Value = 3 Indicates not to publish the Trade to CTM. No trade reports will go to CTM-R and OCC and last sale data will not go to COPP and CFN. Used by One Chicago for block trade functionality.
TradeReportHandlingInstructions.REGULAR	Initial Value = 0 Default value. Used by One Chicago for Block Trade functionality. Indicates the current default functionality.
TradingSessionMethod.OPENOUTCRY	Initial Value = 2
TradingSessionMethod.SBT	Initial Value = 1
TradingSessionStates.CLOSED	Initial Value = 1
TradingSessionStates.OPEN	Initial Value = 2
TradingSessionType.DAY	Initial Value = 1
TradingSessionType.EVENING	Initial Value = 2
UserRoles.BROKER_DEALER	Initial Value = 'B'
UserRoles.CLASS_DISPLAY	Initial Value = 'C'
UserRoles.CUSTOMER_BROKER_DEALER	Initial Value = 'X'
UserRoles.DPM_ROLE	Initial Value = 'D'

UserRoles.EXCHANGE_BROKER	Initial Value = 'E'
UserRoles.FIRM	Initial Value = 'F'
UserRoles.FIRM_DISPLAY	Initial Value = 'R'
UserRoles.HELP_DESK	Initial Value = 'H'
UserRoles.MARKET_MAKER	Initial Value = 'M'
UserRoles.PRODUCT_MAINTENANCE	Initial Value = 'P'
UserRoles.TFL_ROLE	Initial Value = 'T' User role for a Trading Floor Liaison.
UserRoles.UNKNOWN_ROLE	Initial Value = 'K'
VolumeTypes.AON	Initial Value = 2 All or none
VolumeTypes.CUSTOMER_ORDER	Initial Value = 6 Customer
VolumeTypes.FOK	Initial Value = 3 Fill or Kill
VolumeTypes.IOC	Initial Value = 4 Immediate or Cancel
VolumeTypes.LIMIT	Initial Value = 1 Limit (no contingency)
VolumeTypes.NO_CONTINGENCY	Initial Value = 5
VolumeTypes.ODD_LOT	Initial Value = 9 All non-round lot amounts
VolumeTypes.PROFESSIONAL_ORDER	Initial Value = 7 Broker Dealer
VolumeTypes.QUOTES	Initial Value = 8 In-crowd Market-Maker.

Operations

Administrator.sendMessage	Send a message to the CBOE. This will be required so that the client can respond to a message from the CBOE Help Desk. This method is required for the Market-Maker, DPM, Broker and Firm roles. Optional for the Firm Display role. Not allowed for the Market Data role.
CMIClassStatusConsumer.acceptClassState	Accept a sequence of class state changes. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIClassStatusConsumer.updateProductClass	Accept an updated product class struct. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMICurrentMarketConsumer.acceptCurrentMarket	Accepts Current Market information for the client that is published by the client application server. This method is required if either subscribeCurrentMarketForClass or subscribeCurrentMarketForProduct is invoked.
CMIExpectedOpeningPriceConsumer.acceptExpectedOpeningPrice	Accepts Expected Opening Price information for the client. Required if the subscribeExpectedOpeningPrice method is invoked.
CMIIntermarketOrderStatusConsumer.acceptCancelHeldOrderRequest	The NBBO agent accepts the held order cancel request from the customer. This method is used for Intermarket Linkage users only.
CMIIntermarketOrderStatusConsumer.acceptFillRejectReport	The NBBO agent uses this method to receive a report after the NBBO agent rejected a fill report via IntermarketManualHandling::rejectFill. This method is used for Intermarket Linkage users only.
CMIIntermarketOrderStatusConsumer.acceptHeldOrderCanceledReport	The NBBO agent receives a report for a cancel response sent by the NBBO agent via IntermarketManualHandling::acceptCancelResponse This method is used for Intermarket Linkage users only.
CMIIntermarketOrderStatusConsumer.acceptHeldOrderFilledReport	The NBBO agent uses this method to receive a report after the NBBO agent fills the held order. This method is used for Intermarket Linkage users only.
CMIIntermarketOrderStatusConsumer.acceptHeldOrderStatus	The NBBO agent receives the held order status update. This method is used for Intermarket Linkage users only.
CMIIntermarketOrderStatusConsumer.acceptNewHeldOrder	The NBBO agent receives the S order or held order for manual handling. This method is used for Intermarket Linkage users only.
CMINBBOAgentSessionAdmin.acceptBroadcastIntermarketAdminMessage	The NBBO uses this method to receive broadcasted administrative messages. This method is used for Intermarket Linkage users only.
CMINBBOAgentSessionAdmin.acceptForcedOut	The NBBO agent uses this method to receive a forced logout message when another DPM wants to takeover. This method is used for Intermarket Linkage users only.

CMINBBOAgentSessionAdmin.acceptIntermarketAdminMessage	The NBBO uses this method to receive administrative messages. This message is used for the stock opening. This method is used for Intermarket Linkage users only.
CMINBBOAgentSessionAdmin.acceptReminder	Accepts reminder message. This method is used for Intermarket Linkage users only.
CMINBBOAgentSessionAdmin.acceptSatisfactionAlert	Accepts the satisfaction order alert and returns the alert message, class key and trading session to the NBBO agent. This method is used for Intermarket Linkage users only.
CMINBBOConsumer.acceptNBBO	Accept a sequence of NBBO updates. This method is required if either the subscribeNBBOForClass or subscribeNBBOForProduct methods are invoked.
CMIOrderBookConsumer.acceptBookDepth	Accepts book depth information. This method is required if the subscribeBookDepth method is invoked.
CMIOrderBookUpdateConsumer.acceptBookDepthUpdate	Accepts updated book depth information. This method is required if the subscribeBookDepthUpdate method is invoked.
CMIOrderStatusConsumer.acceptNewOrder	Accept a report on a new order entered into the system from the CAS. This method is GMD. It is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderBustReinstateReport	Accept a bust reinstate report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderBustReport	Accept an order bust report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderCanceledReport	Accept order cancel reports from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderFilledReport	Accept an order filled report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role..
CMIOrderStatusConsumer.acceptOrderStatus	Accept a sequence of order detail information. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIProductStatusConsumer.acceptProductState	Accepts a sequence of product state changes. Required for Market Makers, DPMs, Brokers, Firm, and Class Display roles. Optional for the Firm Display role.

CMIProductStatusConsumer.updatedProduct	Accepts an update to a product or a new product created through an intra-day product addition. Required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIQuoteStatusConsumer.acceptQuoteBustReport	Accept a bust trade report for a quote from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIQuoteStatusConsumer.acceptQuoteCancelReport	Accept a cancel report for a quote from the CAS.
CMIQuoteStatusConsumer.acceptQuoteFilledReport	Accept a fill report on a quote from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIQuoteStatusConsumer.acceptQuoteStatus	Accepts quote status from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIRecapConsumer.acceptRecap	Accept a sequence of recap records. This method is required if the subscribeRecapForClass or subscribeRecapForProduct method is invoked.
CMIRFQConsumer.acceptRFQ	Accept a request for quotes. This method is required for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. Required for Futures, optional for Options.
CMIStrategyStatusConsumer.updateProductStrategy	Accepts reports on updates to a sequence of 1 or more strategies. Also accepts new strategy products. Optional for all roles but required for any role that intends to obtain strategy products.
CMITickerConsumer.acceptTicker	Accept a sequence of ticker information. This method is required if the subscribeTicker method is invoked.
CMITradingSessionStatusConsumer.acceptTradingSessionState	Accept reports on changes to trading session state. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIUserSessionAdmin.acceptAuthenticationNotice	The CAS invokes this operation to inform the client application that it is time for the client to re-authenticate with the CAS. If the client application fails to respond to this request from the CAS by invoking cmi::UserSessionManager::authenticate() - the client application will be forcibly logged off by the CAS. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

CMIUserSessionAdmin.acceptCallbackRemoval	Invoked by the CAS to inform the client of callback objects that have been deregistered by the CAS. This normally results when there is a problem in the client callback and a response is not received by the CAS. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIUserSessionAdmin.acceptHeartBeat	The acceptHeartBeat() operation is invoked by the CAS during periods of inactivity to ensure the client application program is still responsive. The CAS expects a returned HeartBeatStruct. This operation can be used to make calls out to local management systems at the client site to, in effect, monitor the CAS. This could be done by contacting an SNMP management system or an application management system. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIUserSessionAdmin.acceptLogout	The CAS uses this operation to inform the client that it has been logged off of the CAS. A reason is provided as a string to this operation. The CAS client application only has to do its own local cleanup - the application is already logged off from the CAS when this callback is invoked. This operation is provided for the convenience of interactive client applications so that the application can inform the user when the CAS is logging the user off of the system. It is assumed that the reason string will be displayed to the user. If logging is being provided in the client application - it is a good idea to record the logout event and reason. As with the acceptHeartBeatI() operation, an interface to a management system at the client location be implemented within this operation for applications, especially non-interactive applications. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
CMIUserSessionAdmin.acceptTextMessage	Accepts text message from the CAS. It will primarily be used for communication from exchange officials to traders. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
IntermarketManualHandling.acceptAdminMessage	Accepts general admin messages from ITS. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptCancelResponse	The NBBO agent responds to a cancel request that originated from the customer. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptCustomerOrderSatisfy	Fill customer order against outgoing S order. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptFillReject	Reject the outbound S order fill report. This method is used for Intermarket Linkage users only.

IntermarketManualHandling.acceptHeldOrderByClassReroute	The NBBO agent reroutes all held orders for a given class. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptHeldOrderFill	The NBBO agent fills the held order. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptHeldOrderReroute	The NBBO agent reroutes the held order back to the system. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptOpeningPriceForProduct	This method is used for stock products only. It allows the NBBO agent to enter the opening price for an equity. Any order imbalance at this price will be routed to manual handling. This method is only available for use on CBOE non-primary equities.
IntermarketManualHandling.acceptPreOpeningIndication	Accept the pre-opening indication message for a security. This method is used by the Stock Intermarket Trading System (ITS).
IntermarketManualHandling.acceptPreOpeningResponse	Accepts the pre-opening response from a DPM participant in Stock ITS.
IntermarketManualHandling.acceptSatisfactionOrderFill	Accepts the incoming Satisfaction order fill. The NBBO agent submits a cross order to partially fill the S order. The NBBO agent allocates some quantity to be distributed among the Market Makers. The NBBO agent also indicates if the remaining quantity should be cancelled. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptSatisfactionOrderInCrowdFill	Accepts incoming Satisfaction order fill. The NBBO agent distributes the quantity among the Market Makers. The NBBO agent indicates if the remaining quantity should be cancelled. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.acceptSatisfactionOrderReject	Rejects the incoming Satisfaction order. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.getAssociatedOrders	The NBBO agent retrieves the related P/A order for a customer order. If the orderId is for a customer order, this method will return the linked P/A orders or the S order. If the orderId is for a Linkage Order (P/A or S orders) this method will return associated customer orders. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.getHeldOrderById	The NBBO agent retrieves the held order by the given order Id. This method is used for Intermarket Linkage users only.
IntermarketManualHandling.getOrdersByOrderTypeAndClass	The NBBO agent retrieves the summary report for P/A, P and S orders (inbound and outbound). This method is used for Intermarket Linkage users only.
IntermarketManualHandling.getOrdersByOrderTypeAndProduct	The NBBO agent retrieves summary report for P/A, P and S orders (inbound and outbound) by product. This method is used for Intermarket Linkage users only.

IntermarketManualHandling.lockProduct	This method is for stock products only. It allows the NBBO Agent to lock the order book at a specified time before the open and up to the opening of a CBOE non-primary equity. This method can only be used before the open.
IntermarketManualHandling.rerouteBookedOrderToHeldOder	This method is for stock products only. It allows the NBBO Agent to remove an order in the order book and place it into manual handling. This method is available before the open of a CBOE non-primary equity.
IntermarketManualHandling.unlockProduct	This method is for stock products only. It allows the NBBO Agent to unlock the order book and let queued orders enter the order book before the open. This is generally used if the primary opening is delayed. This is for CBOE non-primary equities that have not yet opened.
IntermarketQuery.getAdminMessage	Retrieves the admin messages from the intermarket exchanges. This method is used for Intermarket Linkage users only.
IntermarketQuery.getDetailedOrderBook	Retrieves order book details by session and product key. This method is used for Intermarket Linkage users only.
IntermarketQuery.getIntermarketByClassForSession	Retrieves the classes that are listed on more than one exchange for the session. This method is used for Intermarket Linkage users only.
IntermarketQuery.getIntermarketByProductForSession	Retrieves the products that are listed on more than one exchange for the session. This method is used for Intermarket Linkage users only.
IntermarketQuery.getOrderBookStatus	This method will get the state of the OrderBook during the Product Opening state. This call will only be available to the DPM.
IntermarketQuery.showMarketableOrderBookAtPrice	Currently not implemented. This method will get the orders on both the sell side and buy side for the given price. It will also return the buy orders greater than equal to and sell side less than equal to this price. It will also return any Market Orders that are in the OrderBook before the opening. This call will be only available to the DPM.
IntermarketUserAccess.getIntermarketUserSessionManager	Obtain the reference to the Intermarket User Session Manager Service. This method is used for Intermarket Linkage users only.
IntermarketUserAccess.logon	Logon to the CAS. This method is used for Intermarket Linkage users only.
IntermarketUserSessionManager.getIntermarketQuery	Obtain the reference to the Intermarket Query Service. This method is used for Intermarket Linkage users only.
IntermarketUserSessionManager.getNBBOAgent	The CAS returns information about the NBBO agent. This method is used for Intermarket Linkage users only.

MarketQuery.getBookDepth	Retrieve the book depth for a product. This method is not allowed for the Class Display role but it is optional for the Market Maker, DPM, Broker, Firm Display and Firm roles.
MarketQuery.getMarketDataHistoryByTime	Retrieve market data for a specific product starting at a specific time in the past. This method is not allowed for the Class Display role but it is optional for the Market Maker, DPM, Broker, Firm Display and Firm roles.
MarketQuery.subscribeBookDepth	Subscribe for the book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Firm Display and Firm) but not allowed for the Class Display role. This method is not allowed for new development.
MarketQuery.subscribeBookDepthUpdate	Subscribe for the updated book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Firm Display and Firm) but not allowed for the Class Display role. This method is not allowed for new development.
MarketQuery.subscribeCurrentMarketForClass	Subscribe for current market messages for all products belonging to a single class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.subscribeCurrentMarketForProduct	Subscribe the caller to current market data for an individual product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.subscribeExpectedOpeningPrice	Used during opening rotation, this operation subscribes your application to receive the expected opening price for series in a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).This method is not allowed for new development.
MarketQuery.subscribeNBBOForClass	Subscribe for the NBBO for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.subscribeNBBOForProduct	Subscribe for the NBBO for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.subscribeRecapForClass	Subscribe the caller to market data updates for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.

MarketQuery.subscribeRecapForProduct	Subscribe the caller to market data updates for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.subscribeTicker	Subscribe to ticker information for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeBookDepth	Unsubscribe for the book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeBookDepthUpdate	Unsubscribe for the updated book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeCurrentMarketForClass	Unsubscribe for updates to the current market for products belonging to a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeCurrentMarketForProduct	Unsubscribe from market for a specific product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeExpectedOpeningPrice	Unsubscribe opening price data for a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeNBBOForClass	Unsubscribe for the NBBO for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeNBBOForProduct	Unscubscribe for the NBBO for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeRecapForClass	Unsubscribe from recap for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
MarketQuery.unsubscribeRecapForProduct	Unsubscribe from recap for a specific product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.

MarketQuery.unsubscribeTicker	Unsubscribe to ticker information for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm). This method is not allowed for new development.
NBBOAgent.registerAgent	Registers the NBBO agent to the CAS. This method is used for Intermarket Linkage users only.
NBBOAgent.unregisterAgent	Unregisters the NBBO agent from the CAS. This method is used for Intermarket Linkage users only.
NBBOAgentSessionManager.getIntermarketHeldOrderQuery	Obtain the reference to the Intermarket Held Order Query service.
NBBOAgentSessionManager.getIntermarketManualHandling	Obtain the reference to the Intermarket Manual Handling service. This method is used for Intermarket Linkage users only.
OrderEntry.acceptCrossingOrder	Used to submit paired cross orders. The user sends in two orders (Buy and Sell) in the same message. CROSS: Both order will have contingency CROSS the price is the Limit price. MIDPOINT CROSS: Both Order will have contingency MIDPOINT CROSS and the price must be Market price. AUTOLINK the opposite side must be AUTOLINK CROSS MATCH
OrderEntry.acceptOrder	Submit an order. This method is required for Broker and Firm roles if acceptOrderByProductName is not used. Required for Market Maker and DPM roles who intend to submit orders if acceptOrderByProductName is not used. Not allowed for the Class Display or Firm Display role. For Cross orders, the user submits two separate order messages. CROSS: Price is limit price MIDPOINT CROSS: Price is Market Price AUTOLINK CROSS: Price is Limit price AUTOLINK CROSS MATCH
OrderEntry.acceptOrderByProductName	Convenience operation that provides the ability to populate a ProductNameStruct and an OrderEntryStruct and submit it to the CAS for processing. Eliminates the need to perform an initial product key lookup before submitting an order. This method is required for Broker and Firm roles if acceptOrder is not used. Required for Market Maker and DPM roles who intend to submit orders if acceptOrder is not used. Not allowed for the Firm Display or Class Display role.
OrderEntry.acceptOrderCancelReplaceRequest	Request cancellation and replacement of part or all of an order. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for the Firm Display or Class Display role.

OrderEntry.acceptOrderCancelRequest	Request cancellation of part or all of an order. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for the Firm Display or Class Display roles.
OrderEntry.acceptOrderUpdateRequest	Request update of part or all of an order. This method is strongly encouraged for Broker and Firm roles. Strongly encouraged for Market-Makers and DPMs who intend to submit orders. Not allowed for the Firm Display or Class Display roles.
OrderEntry.acceptRequestForQuote	Submit a request for quotation ("RFQ") for a product. This method is required for Broker and Firm roles. Not allowed for the Firm Display or Class Display role. Encouraged for Market-Makers and DPMs.
OrderEntry.acceptStrategyOrder	Submit a strategy (spread, combination, etc.) order. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display roles.
OrderEntry.acceptStrategyOrderCancelReplaceRequest	Request cancellation and replacement of part or all of a strategy (spread, combination, etc.) order. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display roles.
OrderEntry.acceptStrategyOrderUpdateRequest	Update an existing strategy (spread, combination, etc) order. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display roles.
OrderQuery.getOrderById	Retrieve a specific orders. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.
OrderQuery.getOrdersForClass	Retrieve orders of a specific product class. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.
OrderQuery.getOrdersForProduct	Retrieve orders by product for one particular user. This method is encouraged for Broker and Firm roles. Encouraged for Market-Makers and DPMs who intend to submit orders. Not allowed for the Firm Display or Class Display role.
OrderQuery.getOrdersForSession	Obtain orders for a specific trading session and subscribe for subsequent updates to order status for those orders. This method is encouraged for Broker and Firm roles. Encouraged for Market-Makers and DPMs who intend to submit orders. Not allowed for the Firm Display or Class Display role.
OrderQuery.getOrdersForType	Retrieve orders of a specific product type and subscribe client to subsequent updates. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.
OrderQuery.getPendingAdjustmentOrdersByClass	Retrieve pending orders by product class. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.

OrderQuery.getPendingAdjustmentOrdersByProduct	Retrieve pending adjustment orders by product. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.
OrderQuery.queryOrderHistory	Retrieve order history for a specific order. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role.
OrderQuery.subscribeOrders	Subscribe client to receive all orders' status. All current orders for the user will be published by the CAS as a result of this subscription. This method is required for Broker and Firm roles if the subscribeOrdersWithoutPublish method is not used. Required for Market-Makers and DPMs who intend to submit orders if the subscribeOrdersWithoutPublish method is not used. Not allowed for the Firm Display or Class Display role. This method is not allowed for new development.
OrderQuery.subscribeOrdersByFirm	Subcribe for orders for a firm. Only the Firm and Firm Display roles may use this method. This method is not allowed for new development.
OrderQuery.subscribeOrdersByFirmWithoutPublish	Subscribe for orders for a firm without publishing. Only the Firm and Firm Display roles may use this method. This method is not allowed for new development.
OrderQuery.subscribeOrdersWithoutPublish	Subscribe client to receive all orders' status without publishing. This method is required for Broker and Firm roles if the subscribeOrders method is not used. Required for Market-Makers and DPMs who intend to submit orders if the subscribeOrders method is not used. Not allowed for the Firm Display or Class Display role. This method is not allowed for new development.
OrderQuery.unsubscribeAllOrderStatusForType	Unsubscribe from receiving order status information for a specific type of product, such as options. This method is optional for Market Makers, DPMs and Brokers but it is not allowed for the Firm or Market Data roles. This method is not allowed for new development.
OrderQuery.unsubscribeOrderStatusByClass	Unsubscribe from receiving order status information for a specific class. This method is optional for Market Makers, DPMs and Brokers but it is not allowed for the Firm or Market Data roles. This method is not allowed for new development.
OrderQuery.unsubscribeOrderStatusForFirm	Unsubscribe from receiving order status for orders belonging to a firm. Only the Firm and Firm Display roles may use this method. This method is not allowed for new development.

OrderQuery.unsubscribeOrderStatusForProduct	Unsubscribe for order status updates for a specific product. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role. This method is not allowed for new development.
OrderQuery.unsubscribeOrderStatusForSession	Unsubscribe for order status updates for a specific trading session. This method is optional for Broker, Firm, Market-Maker, and DPM roles. Not allowed for Firm Display or Class Display role. This method is not allowed for new development.
ProductDefinition.acceptStrategy	Accept a request to define a strategy (spread, combination, etc.) for trading. System will define a new strategy if one does not already exist. A StrategyStruct is returned that contains the product key for the strategy requested. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductDefinition.buildStrategyRequestByName	Define a strategy (spread, combination, etc.) by name for trading. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductDefinition.buildStrategyRequestByProductKey	Define a strategy (spread, combination, etc.) by product for trading. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getAllPendingAdjustments	Returns sequence of zero or more pending adjustments. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getClassByKey	Retrieve Product Class information by Class Key. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getClassBySymbol	Get information for a class using the class symbol. The product type must be specified. If the third argument is set to true, then the class will only be returned if it is active on the system. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getPendingAdjustmentProducts	Return product name information for products with pending adjustments. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getPendingAdjustments	Return pending adjustment for specific classes. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getProductByKey	Get Product Information by Product Key. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

ProductQuery.getProductByName	Get full product information by name. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getProductClasses	Retrieve the product classes for a product type. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getProductNameStruct	Get full product information by product key. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getProductsByClass	Retrieve all product information for the given class identifier of the given type and subscribe the client to future product updates. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getProductTypes	Retrieve the product type definitions. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getStrategiesByClass	Get a list of strategies for a given class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getStrategiesByComponent	Return a list of strategies that contain the product, designated in the ProductKey argument, as a component (or leg) of the strategy. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.getStrategyByKey	Return the information on a strategy based upon the strategy product key provided as an argument. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
ProductQuery.isValidProductName	Check to see if the given product name is valid. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
Quote.acceptQuote	Submit single quotations. This method is required for Market-Makers and DPMs if the acceptQuotesForClass method is not used. Optional for Market-Makers and DPMs if the acceptQuotesForClass method is used. Not allowed for the Broker, Firm, Firm Display, and Class Display roles.
Quote.acceptQuotesForClass	Submit quotes for multiple products (e.g. Option series) belonging to a single class. This method is required for Market-Makers and DPMs using an autoquoting application that would submit multiple quotes at one time. When autoquoting takes place each method call must contain multiple quotes. If only manual quoting is anticipated, then this method is optional. If not used, then the acceptQuote method is required for Market-Makers and DPMs. Not allowed for the Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.

Quote.cancelAllQuotes	Cancel all current quotations for the client. This method is strongly encouraged for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.
Quote.cancelQuote	Cancel the current quotation for a product. This method is required for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display roles.
Quote.cancelQuotesByClass	Cancel the current quotations for a product class. This method is required for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display roles.
Quote.getQuote	Get quotes for a product. This method is encouraged for Market Makers and DPMs but not allowed for the Broker, Firm and Class Display roles. This method is not allowed for new development.
Quote.subscribeQuoteStatus	Subscribe for quote status. This method is required for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.
Quote.subscribeQuoteStatusForFirm	Subscribe for quote filled reports for specific member firm. Only the Firm and Firm Display roles may use this method. This method is not allowed for new development.
Quote.subscribeQuoteStatusForFirmWithoutPublish	Subscribe for quote status for a firm without publishing the user's current quotes. This method is required for Market Makers and DPMs but not allowed for Broker, Firm, Firm Display and Class Display roles. This method is not allowed for new development.
Quote.subscribeQuoteStatusWithoutPublish	Subscribe for quote status without publishing. This method is optional for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.
Quote.subscribeRFQ	Subscribe client for request for quotes for a class. This method is required for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.
Quote.unsubscribeQuoteStatus	Unsubscribe from receiving any quote status information. This method is optional for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.

Quote.unsubscribeQuoteStatusForFirm	Unsubscribe from receiving quote status information for a firm. Only the Firm and Firm Display roles may use this method. This method is not allowed for new development.
Quote.unsubscribeRFQ	Unsubscribe client from request for quotes for a class. This method is optional for Market-Makers and DPMs but not allowed for Broker, Firm, Firm Display, and Class Display roles. This method is not allowed for new development.
TradingSession.getClassBySessionForKey	Retrieve a list of classes for a session by class key. Optional for all roles
TradingSession.getClassBySessionForSymbol	Retrieve a SessionClassDetailStruct by ticker symbol. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getClassesForSession	Get list of classes for a session and subscribe for updates to classes or class status. (Note: At this time status is maintained and updated at the product level not the class level - you are encouraged to implement your application to accept and apply updates to class, in the event this interface is used in the future). This method is encouraged for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getCurrentTradingSessions	Retrieve current trading sessions and subscribes the user for subsequent trading session updates. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getProductBySessionForKey	Retrieve SessionProductStruct for a specific product by its CBOEdirect product key for a specific trading session. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getProductBySessionForName	Retrieve a SessionProductStruct by its CBOEdirect product key for a specific trading session. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getProductsForSession	Get products for a session for a specific class. Also subscribes the user to receive text messages by class. This method is required for the Market Maker, DPM, Broker and Firm roles but it is encouraged for the Class Display and Firm Display roles.
TradingSession.getProductTypesForSession	Get list of product types that are traded during a specific trading session. This method is encouraged for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getStrategiesByClassForSession	Get strategies that have been defined for trading within a specific trading session and subscribe for subsequent updates. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

TradingSession.getStrategiesByComponent	Supply a product key for a leg and it will return all the strategies that use that product as a leg. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.getStrategyBySessionForKey	Retrieve a SessionStrategyStruct by its CBOEdirect product key for a specific trading session. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.unsubscribeClassesByTypeForSession	Unsubscribe for class updates for a specific product type, such as options. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.unsubscribeProductsByClassForSession	Unsubscribe for updates to product for a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.unsubscribeStrategiesByClassForSession	Unsubcribe for updates to strategies for a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
TradingSession.unsubscribeTradingSessionStatus	Unsubscribe the user from receiving trading session status update messages. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserAccess.logon	Logon to CAS. A successful logon results in the return of a reference to the UserSessionManager for the login session. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserHistory.getTraderClassActivityByTime	Get trader activity for a specific class for a specific time range (start through end time). This method is not allowed for the Firm Display and Class Display role. Optional for all other roles.
UserHistory.getTraderProductActivityByTime	Get trader activity for a product for a specific time (start through end time). This method is not allowed for the Firm Display and Class Display role. Optional for all other roles.
UserPreferenceQuery.getAllSystemPreferences	Returns all system preferences. This method is not allowed for all roles. For CBOE internal use only.
UserPreferenceQuery.getAllUserPreferences	Returns all user preferences for the current user. This method is not allowed for all roles. For CBOE internal use only.
UserPreferenceQuery.getSystemPreferencesByPrefix	Gets system preferences that match the prefix provided as an argument. This method is not allowed for all roles. For CBOE internal use only.
UserPreferenceQuery.getUserPreferencesByPrefix	Returns user preferences that match the prefix specified in the retrieval list. This method is not allowed for all roles. For CBOE internal use only.

UserPreferenceQuery.removeUserPreference	Removes user preferences specified as a list of preferences. This method is not allowed for all roles. For CBOE internal use only.
UserPreferenceQuery.removeUserPreferencesByPrefix	Removes user preferences that match the prefix specified as an argument. This method is not allowed for all roles. For CBOE internal use only.
UserPreferenceQuery.setUserPreferences	Persists user preferences within the User Preference Service. This method is not allowed for all roles. For CBOE internal use only.
UserSessionManager.authenticate	Used to re-authenticate the client to the CAS after a period of inactivity is detected by the CAS. This method is invoked as a response to an invocation of the UserSessionAdminCallback.acceptAuthenticationNotice() operation. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserSessionManager.changePassword	This operation is provided to change the password for the user currently logged into the CAS. You must specify the old password and the new password. Developers of interactive applications are strongly encouraged to prompt the user to confirm the new password beforehand and then validate the new password by comparing the new password with the confirmed version prior to invoking the changePassword() operation. This method is required in case the CBOE requires users to change their password at some point in time for security reasons. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserSessionManager.getAdministrator	Obtain a reference to the CAS administrator object. This method is required for the Market Maker, DPM, Broker and Firm roles. Optional for the Firm Display role. Not allowed for the Class Display role.
UserSessionManager.getMarketQuery	Obtain the reference to the Market Query Service. This method is optional for Market Maker, DPM, Broker, Firm Display and Firm roles but required for the Class Display role.
UserSessionManager.getOrderEntry	Obtain the reference to the Order Entry service. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for the Class Display and Firm Display roles.
UserSessionManager.getOrderQuery	Obtain the reference to the Order Query Service. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for the Class Display and Firm Display roles.
UserSessionManager.getProductDefinition	Obtain the reference to the Product Definition Service. This method is implemented but not used in Version 1.0 of CBOEdirect.
UserSessionManager.getProductQuery	Obtain the reference to the Product Query Service. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

UserSessionManager.getQuote	Obtain the reference to the Quote Service. This method is required for Market Makers and DPMs, optional for the Broker Dealer role. Not allowed for Class Display role. Optional for the Firm and Firm Display roles.
UserSessionManager.getSystemDateTime	Returns the System Date and Time from the CAS. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserSessionManager.getTradingSession	Obtain a reference to the Trading Session Service. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserSessionManager.getUserHistory	Get the reference to the user history object that is used to retrieve trader history. This method is not allowed for the Class Display or Firm Display roles. It is optional for the Market Maker, DPM, Firm and Broker roles.
UserSessionManager.getUserPreferenceQuery	Obtains the reference to the User Preference Service. This method is not allowed for all roles. For CBOE internal use only.
UserSessionManager.getUserTradingParameters	Returns the UserTradingParameters interface on the CAS. This method is not allowed for the Class Display or Firm Display roles. It is optional for Market Maker, DPM, Broker and Firm roles.
UserSessionManager.getValidSessionProfileUser	Gets the valid session profile for the user from the session profile user struct. A user can have different profiles for different sessions.
UserSessionManager.getValidUser	The CAS returns detailed information about the user (such as clearing firm, Q-account, etc.) currently logged into the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for Class Display and Firm Display roles.
UserSessionManager.getVersion	Obtain the reference to the Version Service. A VersionLable is returned from the CAS. This is crucial to match the version of CMi that the client is using with the version that the CBOE is using. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserSessionManager.logout	Logs the current user off of the CAS. This method is required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
UserTradingParameters.getAllQuoteRiskProfiles	Return the quote risk management profile for the current user. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.getDefaultQuoteRiskProfile	Get the default quote risk management profile. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.

UserTradingParameters.getQuoteRiskManagementEnabledStatus	Get the quote risk management enabled status. Returns true if quote risk management is enabled. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.getQuoteRiskManagementProfileByClass	Return the quote risk management profile for a specific class. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.removeAllQuoteRiskProfiles	Remove all quote risk management profiles. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.removeQuoteRiskProfile	Remove a quote risk management profile for a class. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.setQuoteRiskManagementEnabledStatus	Turn quote risk management on or off. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
UserTradingParameters.setQuoteRiskProfile	Set a quote risk management profile for a class. This method is optional for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display role.
Version.getVersionNumber	Returns the version number as type Version of the IDL that was used to generate the application. Required for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

CMi V2 Operations

CMICurrentMarketConsumer.acceptCurrentMarket	Accepts Current Market information for the client that is published by the client application server. This method is required if either subscribeCurrentMarketForClass or subscribeCurrentMarketForProduct is invoked.
CMIExpectedOpeningPriceConsumer.acceptExpectedOpeningPrice	Accepts Expected Opening Price information for the client. Required if the subscribeExpectedOpeningPrice method is invoked.
CMILockedQuoteStatusConsumer.acceptQuoteLockedReport	Accept a locked report on a quote from the CAS.
CMINBBOConsumer.acceptNBBO	Accepts the National Best Bid and Offer.
CMIOrderBookConsumer.acceptBookDepth	Accepts book depth information. This method is required if the subscribeBookDepth method is invoked.
CMIOrderBookUpdateConsumer.acceptBookDepthUpdate	Accepts updated book depth information. This method is required if the subscribeBookDepthUpdate method is invoked.
CMIOrderStatusConsumer.acceptNewOrder	Accept a report on a new order entered into the system from the CAS. This method is GMD. It is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderBustReinstateReport	Accept a bust reinstate report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderBustReport	Accept an order bust report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderCanceledReport	Accept order cancel reports from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderFilledReport	Accept an order filled report from the CAS. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.
CMIOrderStatusConsumer.acceptOrderStatus	Accept a sequence of order detail information. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Optional for the Firm Display role. Not allowed for the Class Display role.

CMIQuoteStatusConsumer.acceptQuoteBustReport	Accept a bust trade report for a quote from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIQuoteStatusConsumer.acceptQuoteDeleteReport	Accepts a delete report for a quote from the CAS.
CMIQuoteStatusConsumer.acceptQuoteFillReport	Accept a fill report on a quote from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIQuoteStatusConsumer.acceptQuoteStatus	Accepts quote status from the CAS. This method is required for Market-Makers and DPMs. Optional for Firm and Firm Display roles. Not allowed for the Broker or Class Display roles.
CMIRecapConsumer.acceptRecap	Accept a sequence of recap records. This method is required if the subscribeRecapForClass or subscribeRecapForProduct method is invoked.
CMIRFQConsumer.acceptRFQ	Accepts the request for quote.
CMITickerConsumer.acceptTicker	Accept a sequence of ticker information. This method is required if the subscribeTicker method is invoked.
MarketQuery.getBookDepthDetails	Retrieves book depth details. Optional for all roles. Not allowed for any class in W_MAIN. Not allowed for new development.
MarketQuery.subscribeBookDepthForClassV2	Subscribe for the book depth of a class. This method is optional for all roles but required if the firm wishes to subscribe to Book Depth, unless subscribeBookDepthForProductV2 is used. Not allowed for any class in W_MAIN. Not allowed for new development.
MarketQuery.subscribeBookDepthForProductV2	Subscribe for the book depth of a product. This method is optional for all roles but required if the firm wishes to subscribe to Book Depth Updates, unless subscribeBookDepthUpdateForProductV2 is used. Not allowed for any class in W_MAIN. Not allowed for new development.
MarketQuery.subscribeBookDepthUpdateForClassV2	Subscribe for the updated book depth of a class. This method is optional for all roles but required if the firm wishes to subscribe to Book Depth Updates, unless subscribeBookDepthUpdateForProductV2 is used. Not allowed for any class in W_MAIN. Not allowed for new development.
MarketQuery.subscribeBookDepthUpdateForProductV2	Subscribe for the updated book depth of a product. This method is optional for all roles but required if the firm wishes to subscribe to Book Depth Updates, unless subscribeBookDepthUpdateForClassV2 is used. Not allowed for any class in W_MAIN. Not allowed for new development.

MarketQuery.subscribeCurrentMarketForClassV2	Subscribe for current market messages for all products belonging to a single class. This method is optional for all roles, but required if the firm wishes to subscribe to Current Market unless subscribeCurrentMarketForProductV2 is used.
MarketQuery.subscribeCurrentMarketForProductV2	Subscribe the caller to current market data for an individual product. This method is optional for all roles but required if the firm wishes to subscribe to Current Market unless subscribeCurrentMarketForClassV2 is used.
MarketQuery.subscribeExpectedOpeningPriceForClassV2	Used during opening rotation, this operation subscribes your application to receive the expected opening price for a class. This method is required for Options DPMs. Optional for all other roles, but required if the firm wishes to subscribe to EOP unless subscribeExpectedOpeningPriceForProductV2 is used.
MarketQuery.subscribeExpectedOpeningPriceForProductV2	Used during opening rotation, this operation subscribes your application to receive the expected opening price for a product. This method is required for Options DPMs. Optional for all other roles, but required if the firm wishes to subscribe to EOP unless subscribeExpectedOpeningPriceForClassV2 is used.
MarketQuery.subscribeNBBOForClassV2	Subscribe for the NBBO for a class. This method is optional for all roles.
MarketQuery.subscribeNBBOForProductV2	Subscribe for the NBBO for a product. This method is optional for all roles.
MarketQuery.subscribeRecapForClassV2	Subscribe the caller to market data updates for a class. This method is optional for all roles but required if the firm wishes to subscribe to Recap unless subscribeRecapForProductV2 is used.
MarketQuery.subscribeRecapForProductV2	Subscribe the caller to market data updates for a product. This method is optional for all roles but required if the firm wishes to subscribe to Recap unless subscribeRecapForClassV2 is used.
MarketQuery.subscribeTickerForClassV2	Subscribe to ticker information for a class. This method is optional for all roles but required if the firm wishes to subscribe to Ticker unless subscribeTickerForProductV2 is used.
MarketQuery.subscribeTickerForProductV2	Subscribe to ticker information for a product. This method is optional for all roles but required if the firm wishes to subscribe to Ticker unless subscribeTickerForClassV2 is used.
MarketQuery.unsubscribeBookDepthForClassV2	Unsubscribe for the book depth of a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeBookDepthForProductV2	Unsubscribe for the book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeBookDepthUpdateForClassV2	Unsubscribe for the book depth of a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).

MarketQuery.unsubscribeBookDepthUpdateForProductV2	Unsubscribe for the updated book depth of a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeCurrentMarketForClassV2	Unsubscribe for updates to the current market for products belonging to a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeCurrentMarketForProductV2	Unsubscribe from market for a specific product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeExpectedOpeningPriceForClassV2	Unsubscribe opening price data for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeExpectedOpeningPriceForProductV2	Unsubscribe opening price data for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeNBBOForClassV2	Unsubscribe for the NBBO for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeNBBOForProductV2	Unscubscribe for the NBBO for a product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeRecapForClassV2	Unsubscribe from recap for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeRecapForProductV2	Unsubscribe from recap for a specific product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeTickerForClassV2	Unsubscribe to ticker information for a class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeTickerForProductV2	Unsubscribe to ticker information for a specific product. TThis method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
OrderQuery.subscribeOrderStatusForClassV2	Subscribe for order status information for a specific class. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for the Class Display and Firm Display roles. Optional if the subscribeOrderStatusV2 method is used.
OrderQuery.subscribeOrderStatusForFirmForClassV2	Subscribe for order status by class for orders belonging to a Firm . This method is required for Broker and Firm roles. Not allowed for Market-Maker, DPM, Class Display and Firm Display roles. Optional if the subscribeOrderStatusForFirmV2 method is used.
OrderQuery.subscribeOrderStatusForFirmV2	Subcribe for orders for a firm. This method is required for Broker and Firm roles. Not allowed for Market-Maker, DPM Class Display, and Firm Display roles. Optional if the subscribeOrderStatusForFirmForClassV2 method is used.

OrderQuery.subscribeOrderStatusV2	Subscribe for order status updates. This method is required for Broker and Firm roles. Required for Market-Makers and DPMs who intend to submit orders. Not allowed for Class Display and Firm Display roles. Optional if the subscribeOrderStatusForClassV2 method is used.
OrderQuery.unsubscirbeOrderStatusForClassV2	Unsubscribe from receiving order status information for a specific class. This method is optional for all roles.
OrderQuery.unsubscribeOrderStatusForFirmForClassV2	Unsubscribe from receiving order status by class for orders belonging to a firm. This method is optional for all roles.
OrderQuery.unsubscribeOrderStatusForFirmV2	Unsubscribe from receiving order status for orders belonging to a firm. This method is optional for all roles.
OrderQuery.unsubscribeOrderStatusV2	Unsubscribe for order status updates. This method is optional for all roles.
Quote.acceptQuotesForClassV2	Submit quotes for multiple products (e.g. Option series) belonging to a single class. This method is required for Market-Makers and DPMs using an autoquoting application that would submit multiple quotes at one time. When autoquoting takes place each method call must contain multiple quotes. If only manual quoting is anticipated, then this method is optional. If not used, then the acceptQuote method is required for Market-Makers and DPMs. Not allowed for all other roles.
Quote.subscribeQuoteLockedNotification	Subscribe for quote locked notification. This method is required for Market-Makers and DPMs in Hybrid. Optional for Broker-Dealer role if "I" orders are used. Not allowed for other roles or for Futures. Optional if subscribeQuoteLockedNotificationForClass is used.
Quote.subscribeQuoteLockedNotificationForClass	Subscribe for quote locked notification by class. This method is required for Market-Makers and DPMs in Hybrid. Optional for Broker-Dealer role if "I" orders are used. Not allowed for all other roles or for Futures. Optional if subscribeQuoteLockedNotification is used.
Quote.subscribeQuoteStatusForClassV2	Subscribe for quote status by class. This method is required for Market-Makers and DPMs but not allowed for all other roles. Optional if subscribeQuoteStatusForFirmForClassV2 is used.
Quote.subscribeQuoteStatusForFirmForClassV2	Subscribe for quote filled reports by class for specific member firm. This method is optional for Firm and Firm Display roles.
Quote.subscribeQuoteStatusForFirmV2	Subscribe for quote filled reports for specific member firm. This method is optional for Firm and Firm Display roles.
Quote.subscribeQuoteStatusV2	Subscribe for quote status. This method is required for Market-Makers and DPMs but not allowed other roles. Optional if subscribeQuoteStatusForFirmForClassV2 is used.

Quote.subscribeRFQV2	Subscribe client for request for quotes for a class. This method is required for Futures Market-Makers. Not allowed for Broker, Firm, Firm Display, and Class Display roles.
Quote.unsubscribeQuoteLockedNotification	Unsubscribe for quote locked notification. This method is optional for all roles.
Quote.unsubscribeQuoteLockedNotificationForClass	Unsubscribe for quote locked notification by class. This method is optional for all roles.
Quote.unsubscribeQuoteStatusForClassV2	Unsubscribe from receiving quote status by class. This method is optional for all roles.
Quote.unsubscribeQuoteStatusForFirmForClassV2	Unsubscribe from receiving quote status information by class for a firm. This method is optional for all roles.
Quote.unsubscribeQuoteStatusForFirmV2	Unsubscribe from receiving quote status information for a firm. This method is optional for all roles.
Quote.unsubscribeQuoteStatusV2	Unsubscribe from receiving any quote status information. This method is optional for all roles.
Quote.unsubscribeRFQV2	Unsubscribe client from request for quotes for a class. This method is optional for all roles.
UserAccessV2.getUserSessionManagerV2	Retrieve information from the UserSessionManager. This method is required for Hybrid and all V2.5 users.
UserAccessV2.logon	Logon to CAS. A successful logon results in the return of a reference to the UserSessionManager for the login session. This method is required for Hybrid and V2.5 users.
UserSessionManagerV2.getMarketQueryV2	Obtain the reference to the Market Query Service. This method is optional for Hybrid and all V2.5 users, but must be used if Market Query is used at all.
UserSessionManagerV2.getOrderQueryV2	Obtain the reference to the Order Query Service. This method is optional for all V2.5 users, but must be used if Order Query is used at all.
UserSessionManagerV2.getQuoteV2	Obtain the reference to the Quote Service. This method is required for Market-Makers and DPMs. Optional for all other users, but must be used if Quote messaging is used at all.

CMi V3 Operations

CMIAuctionConsumer.acceptAuction	Accepts auction data.
CMICurrentMarketConsumer.acceptCurrentMarket	Accepts Current Market information for the client that is published by the client application server. This method is required if either subscribeCurrentMarketForClassV3 or subscribeCurrentMarketForProductV3 is invoked.
MarketQuery.getDetailProductHistoryByTime	Retrieves details of the product time.
MarketQuery.getPriorityProductHistoryByTime	Retrieves the product history time by priority.
MarketQuery.subscribeCurrentMarketForClassV3	Subscribe for current market messages for all products belonging to a single class. This method is optional for all roles, but required if the firm wishes to subscribe to Current Market unless subscribeCurrentMarketForProductV3 is used.
MarketQuery.subscribeCurrentMarketForProductV3	Subscribe the caller to current market data for an individual product. This method is optional for all roles but required if the firm wishes to subscribe to Current Market unless subscribeCurrentMarketForClassV3 is used.
MarketQuery.unsubscribeCurrentMarketForClassV3	Unsubscribe for updates to the current market for products belonging to a specific class. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
MarketQuery.unsubscribeCurrentMarketForProductV3	Unsubscribe from market for a specific product. This method is optional for all roles (Market Maker, DPM, Broker, Class Display, Firm Display and Firm).
OrderEntry.acceptInternalizationOrder	This method is used to submit internalization orders. This method must contain a primary (customer) order and a match (firm) order.
OrderQuery.subscribeAuctionForClass	This method is used to subscribe for auction solicitation from CBOE. The method will take a sequence of auction types. If the wrong session or class is submitted, a data validation will be thrown. If an auction type is invalid, a result struct will be generated.
OrderQuery.unsubscribeAuctionForClass	This method is used to unsubscribe from receiving auction solicitation from CBOE. The method will take a sequence of auction types. If the wrong session or class is submitted, a data validation will be thrown. If an auction type is invalid, a result struct will be generated.

Quote.acceptQuotesForClassV3	Submit quotes for multiple products (e.g. Option series) belonging to a single class. This method is required for Market-Makers and DPMs using an autoquoting application that would submit multiple quotes at one time. When autoquoting takes place each method call must contain multiple quotes. If only manual quoting is anticipated, then this method is optional. If not used, then the acceptQuote method is required for Market-Makers and DPMs. Not allowed for all other roles.
Quote.cancelAllQuotesV3	Cancel all current quotations for the client. This method is strongly encouraged for Market-Makers and DPMs but not allowed for the Broker, Firm, Firm Display, and Class Display roles. This method displays a data validation exception when the trading session name is not valid.
UserAccessV3.logon	Logon to CAS. A successful logon results in the return of a reference to the UserSessionManagerV3 for the login session.
UserSessionManagerV3.getMarketQueryV3	Obtain the reference to the Market Query Service.
UserSessionManagerV3.getOrderEntryV3	References the OrderEntry method.
UserSessionManagerV3.getOrderQueryV3	References the OrderQuery method
UserSessionManagerV3.getQuoteV3	Obtain the reference to the Quote Service.

CMi V4 Operations

CMICurrentMarketConsumer.acceptCurrentMarket	Accepts current market information for the client that is published by the client application server. This method is required if the subscribeCurrentMarket method is invoked.
CMIRecapConsumer.acceptLastSale	Accept a sequence of last sale recores. This method is required if the subscribeRecap method is invoked.
CMIRecapConsumer.acceptRecap	Accept a sequence of recap records. This method is required if the subscribeRecap method is invoked.
CMITickerConsumer.acceptTicker	Accept a sequence of ticker information. This method is required if the subscribeTicker method is invoked.
MarketQuery.subscribeCurrentMarket	Subscribe for MDX current market messages for all products belonging to a single class.
MarketQuery.subscribeRecap	Subscribe for MDX recap and last sale messages for all products belonging to a single class
MarketQuery.subscribeTicker	Subscribe for MDX ticker messages for all products belonging to a single class.
MarketQuery.unsubscribeCurrentMarket	Unsubscribe for updates to the current market for all products belonging to a specific class.
MarketQuery.unsubscribeRecap	Unsubscribe for updates to recaps and last sales for all products belonging to a specific class.
MarketQuery.unsubscribeTicker	Unsubscribe for updates to the ticker for a specific class.
UserAccessV4.logon	Logon to CAS. A successful logon results in the return of a reference to the UserSessionManagerV4 for the login session.
UserSessionManagerV4.getMarketQueryV4	Obtain the reference to the cmiV4::MarketQuery Service.

CMi V5 Operations

OrderEntry.acceptInternalizationStrategyOrder	This method is used to submit internalization strategy orders. This method must contain a primary (customer) order and a match (firm) order
Quote.cancelAllQuotesV5	The Quote Service is used to submit and cancel quotes. The cancelAllQuotesV5 method takes an extra boolean parameter to specify whether a cancel report needs to be sent or not. Refer to API-02 for details on V5.0 functionality.
Quote.cancelQuotesByClassV5	The Quote Service is used to submit and cancel quotes. The cancelQuotesByClassV5 method takes an extra boolean parameter to specify whether a cancel report needs to be sent or not. Refer to API-02 for details on V5.0 functionality.
Quote.cancelQuoteV5	The Quote Service is used to submit and cancel quotes. The cancelQuoteV5 method takes an extra boolean parameter to specify whether a cancel report needs to be sent or not. Refer to API-02 for details on V5.0 functionality.
UserAccessV5.logon	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV5 for the login session.
UserSessionManagerV5.getOrderEntryV5	References the OrderEntry method
UserSessionManagerV5.getQuoteV5	References the Quote service.
UserSessionManagerV5.getUserTradingParametersV5	References the UserTradingParameters interface
UserTradingParameters.getUserRateSettings	This method takes a session name and returns the current values of various rate limits. Currently, this method returns the quote rate limit, quote call limit, order rate limit, order call limit and book depth call limit.

CMi V6 Operations

FloorTradeMaintenanceService.acceptFloorTrade	Used to generate a trading floor trade (i.e. Market Maker Hand Held trade).
FloorTradeMaintenanceService.deleteFloorTrade	This method is used to delete a trading floor trade (i.e. Market Maker Hand Held trade).
FloorTradeMaintenanceService.subscribeForFloorTradeReportsByClass	This method is used to subscribe for trading floor trade (i.e. Market Maker Hand Held) notifications by class. If the classKey is set to zero, the user receives trade notifications for all classes. If the classKey has a valid non-zero value, the user gets trade notifications for the specified classKey only. If neither subscription exists, the CAS will drop the subscription.
FloorTradeMaintenanceService.unsubscribeForFloorTradeReportsByClass	This method is used to turn off floor trade notifications (i.e Market Maker Hand Held trades). If the classKey is zero, notification is truned off for all classes. If the classKey has a valid non-zero value, notification is turned off only for the specified classkey.
OrderQuery.registerForDirectedAIM	Used to register for Directed AIM auction participation.
ProductQueryV6.getClassesForProductGroup	Retrieves the classes in the product group.
ProductQueryV6.getProductGroups	Retrieves a group of products.
UserAccessV6.logon	Used to logon to the CAS to access V6 interfaces.
UserSessionManagerV6.getFloorTradeMaintenanceService	Calls the FloorTradeMaintenanceService.
UserSessionManagerV6.getOrderQueryV6	References the OrderQuery interface.

CMi V7 Operations

OrderEntry.acceptCrossingOrderNoAckV7	This method is used to submit cross orders. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptInternalizationOrderNoAckV7	This method is used to submit internalization orders. This method must contain a primary (customer) order and a match (firm) order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptInternalizationStrategyOrderNoAckV7	This method is used to submit internalization strategy orders. This method must contain a primary (customer) order and a match (firm) order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptInternalizationStrategyOrderV7	This method is used to submit internalization strategy orders. This method must contain a primary (customer) order and a match (firm) order. This method returns an asynchronous new order acknowledgement report.
OrderEntry.acceptOrderByProductNameNoAckV7	Provides the ability to populate a ProductNameStruct and an OrderEntryStruct and submit it to the CAS for processing. Eliminates the need to perform an initial product key lookup before submitting an order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptOrderCancelReplaceRequestNoAckV7	Request cancellation and replacement of part or all of an order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptOrderNoAckV7	This method is used to submit orders that do not generate asynchronous new order acknowledgement report.
OrderEntry.acceptStrategyOrderCancelReplaceRequestNoAckV7	Request cancellation and replacement of part or all of a strategy (spread, combination, etc.) order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptStrategyOrderCancelReplaceRequestV7	Request cancellation and replacement of part or all of a strategy (spread, combination, etc.) order. This method returns an asynchronous new order acknowledgement report.
OrderEntry.acceptStrategyOrderNoAckV7	Submit a strategy (spread, combination, etc.) order. This method will not generate asynchronous new order acknowledgement report.
OrderEntry.acceptStrategyOrderV7	Submit a strategy (spread, combination, etc.) order. This method returns an asynchronous new order acknowledgement report.
Quote.acceptQuotesForClassV7	Submit quotes for multiple products (e.g. Option series) belonging to a single class. Returns a sequence of ClassQuoteResultStructV3.
Quote.acceptQuoteV7	Submit single quotations.

UserAccessV7.logon	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV7 for the login session.
UserSessionManagerV7.getOrderEntryV7	Calls the OrderEntry interface.
UserSessionManagerV7.getQuoteV7	Calls the quote interface.

CMi V8 Operations

TradingClassStatusQuery.getClassesForProductGroup	Takes a group name as a parameter and retruns the list classes traded in that group.
TradingClassStatusQuery.getProductGroups	Returns a lilst of all trading groups irrespective of trading session.
TradingClassStatusQuery.productGroupName	Takes the string, ProductGroup, in cmi::Product which contains list of classes traded in the group.
TradingClassStatusQuery.subscribeTradingClassStatusForClasses	This method is used to subscribe at the class level. A class may or may not have products/series defined under them. This method takes in the trading session (i.e, W_MAIN), the list of class keys and a callback to notify end clients.
TradingClassStatusQuery.subscribeTradingClassStatusForProductGroup	This method is used to subscribe at the group level. A group will have at least one or more trading classes. The method takes in the trading session (i.e W_MAIN), the list group name and a callback to notify end clients.
UserAccessV8.logon	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV8 for the login session.
UserSessionManagerV8.getTradingClassStatusQuery	References the interface, TradingClassStatusQuery.

CMi V9 Operations

OrderEntry.acceptLightOrder	This method is used to submit light orders into the system
OrderEntry.acceptLightOrderCancelRequest	Request cancellation of a light order.
OrderEntry.acceptLightOrderCancelRequestById	Request cancellation of a light order by ID.
UserAccessV9.logon	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV9 for the login session. Used to logon to the CAS to access V9 interfaces.
UserSessionManagerV9.getOrderEntryV9	Calls the order entry interface.

CMi V10 Operations

OrderEntry.acceptLightOrderCancelReplaceRequest	This method is used to submit light orders cancel/replace requests into the system
UserAccessV10.logonV2	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV10 for the login session. Used to logon to the CAS to access V10 interfaces.
UserAccessV10.logon	Logon to the CAS. A successful logon results in the return of a reference to the UserSessionManagerV10 for the login session. Used to logon to the CAS to access V10 interfaces.
UserSessionManagerV10.getOrderEntryV10	Calls the order entry interface.
UserSessionManagerV10.getQuoteV10	Calls the quote interface.
Quote.subscribeQuoteFillStatus	This method is used to subscribe to receive only quote fill messages.
Quote.unsubscribeQuoteFillStatus	This method is used to unsubscribe from receiving only quote fill messages.
Quote.subscribeQuoteFillStatusForClass	This method is used to subscribe to receive only quote fill messages for a class.
Quote.unsubscribeQuoteFillStatusForClass	This method is used to unsubscribe from receiving only quote fill messages for a class.
Quote.subscribeQuoteStatusWithoutFill	This method is used to subscribe to receive quote status messages, except for quote fill messages.
Quote.unsubscribeQuoteStatusWithoutFill	This method is used to unsubscribe from receiving quote status messages without fill messages.
Quote.subscribeQuoteStatusWithoutFillForClass	This method is used to subscribe to receive quote status messages, except for quote fill messages, for a class.
Quote.unsubscribeQuoteStatusWithoutFillForClass	This method is used to unsubscribe from receiving quote status messages without fill messages for a class.

Exceptions

AlreadyExistsException	The information (such as an order) already exists on the server.
AuthenticationException	Unable to login user due to invalid password or login ID.
AuthorizationException	The current user is not authorized to perform the request. An example of this would be a user that is not authorized to receive market data. Many of the MarketQuery requests will fail with an AuthorizationExcepiton.
CommunicationException	Communication between trading system components can not be accomplished making it impossible to satisfy your request. Most likely a server component has failed and a product is no longer able to trade. The recovery mode is to wait until the product is closed and brought back on-line. This error will also be generated if the CAS can not communicate with exchange services.
DataValidationException	The DataValidation Exception is used for any problems with data items in messages sent to the system. This is a recoverable exception and should be caught separately from other exceptions. If your application is interactive this exception should be caught and the error code (cmiErrorCodes.DataValidationCodes) should be interpreted and reported to the user.
NotAcceptedException	The request you have made is not permitted at this time. This may occur if you try to perform a request when a product is not trading.
NotFoundException	The object you have requested (such as a product or class) was not found on the server. This exception should be caught and used to provide an error message back to the user in an interactive application.
NotSupportedException	The request is currently not supported. This exception is used when there is functionality exposed in the CMi API that is not currently supported by the server.
SystemException	A system component has failed. Because the electronic trading system is distributed, it most likely that only a class or product is effected by the outage.
TransactionFailedException	Your transaction request has failed. Refer to the cmiErrorCodes.TransactionFailedCodes for the enumeration of possible failures. Some of these failures are recoverable.

Error Codes

AuthenticationCodes

FUNCTION_NOT_IMPLEMENTED	ErrorCode = 2020
INCORRECT_PASSWORD	ErrorCode = 2010
INVALID_CLIENT_LOGIN_MODE	ErrorCode = 2030
UNKNOWN_USER	ErrorCode = 2000
USER_NOT_ENABLED	ErrorCode = 2040

AuthorizationCodes

AUTHENTICATION_ERROR	ErrorCode = 7005
INVALID_LOCATION	ErrorCode = 7006
INVALID_PASSWORD	ErrorCode = 7004
INVALID_SESSION_ID	ErrorCode = 7002
MAX_TIMEOUT_EXCEEDED	ErrorCode = 7003
NOT_PERMITTED	ErrorCode = 7001
USER_DISABLED	ErrorCode = 7000
USER_NOT_ENABLED_FOR_LIGHT_ORDERS	ErrorCode = 7051

CommunicationFailureCodes

LOST_CONNECTION	ErrorCode = 2520
ROUTING_SESSION_UNAVAILABLE	ErrorCode = 2510
SERVER_NOT_AVAILABLE	ErrorCode = 2530
TIME_OUT	ErrorCode = 2540
TRANSPORT_FAILURE	ErrorCode = 2500

DataValiditionCodes

BUSINESS_DAY_NOT_STARTED	ErrorCode = 1320
CANCELED_VOL_NOT_CUMULATIVE_VOL	ErrorCode = 1807

DUPLICATE_ID	ErrorCode = 1000
DUPLICATE_STRATEGY_LEG	ErrorCode = 1350
EITHER_LAST_SALE_OR_OPENING_TRADE	ErrorCode = 1805
END_OF_SALE	ErrorCode = 1800
GMD_LISTENER_ALREADY_REGISTERED	ErrorCode = 1400
GROUPELEMENT_ALREADY_EXISTS	ErrorCode = 1467
GROUPRELATIONSHI_ALREADY_EXISTS	ErrorCode = 1468
INCOMPLETE_QUOTE	ErrorCode = 1030
INTERNALIZATION_NOT_ALLOWED	ErrorCode = 1630
INVALID_ACCOUNT	ErrorCode = 1200
INVALID_AUCTION_ID	ErrorCode = 1620
INVALID_AUCTION_STATE	ErrorCode = 1610
INVALID_AUCTION_TYPE	ErrorCode = 1640
INVALID_BRANCH_SEQUENCE_NUMBER	ErrorCode = 1300
INVALID_CABINET_ON_CXLRE	ErrorCode = 1905
INVALID_CANCEL_REQUEST	ErrorCode = 1370
INVALID_CLEARING_FIRM	ErrorCode = 1715
INVALID_CONTINGENCY_BOB_IORDER	ErrorCode = 1700
INVALID_CONTINGENCY_TYPE	ErrorCode = 1220
INVALID_CONTINGENCY_VIX_SETTLEMENT	ErrorCode = 1701
INVALID_CONTRA_BROKER	ErrorCode = 1900
INVALID_CONTRA_FIRM	ErrorCode = 1901
INVALID_COUNT_BOUNDARIES	ErrorCode = 1534
INVALID_COVERAGE	ErrorCode = 1260
INVALID_EXCHANGE	ErrorCode = 1500
INVALID_EXECUTING_BROKER	ErrorCode = 1903
INVALID_EXECUTING_BROKER_FIRM	ErrorCode = 1904
INVALID_EXECUTING_GIVEUP_FIRM	ErrorCode = 1210
INVALID_EXTENSIONS	ErrorCode = 1510
INVALID_FIELD_LENGTH	ErrorCode = 1330
INVALID_GROUP	ErrorCode = 1460
INVALID_GROUP_TYPE	ErrorCode = 1473
INVALID_GROUPELEMENT	ErrorCode = 1474
INVALID_GROUPELEMENT_RELATIONSHIP	ErrorCode = 1466

INVALID_GROUPELEMENT_TYPE	ErrorCode = 1465
INVALID_ID	ErrorCode = 1530
INVALID_LEG_CONTINGENCY	ErrorCode = 1360
INVALID_LEG_STOCK_PROD_STATE	ErrorCode = 1714
INVALID_LIMITS	ErrorCode = 1532
INVALID_LOGIN_MODE	ErrorCode = 1390
INVALID_MATCH_TYPE	ErrorCode = 1600
INVALID_NAME	ErrorCode = 1461
INVALID_NO_CURRENT_MARKET	ErrorCode = 1906
INVALID_NO_MATCHING_MDH	ErrorCode = 1908
INVALID_OPENING_REQUIREMENT	ErrorCode = 1440
INVALID_OPERATION_TYPE	ErrorCode = 1475
INVALID_OPERATOR	ErrorCode = 1463
INVALID_ORIGIN_TYPE_FOR_LIGHT_ORDERS	ErrorCode = 7200
INVALID_OPTIONAL_DATA	ErrorCode = 1650
INVALID_ORDER_ID	ErrorCode = 1130
INVALID_ORDER_SOURCE	ErrorCode = 1290
INVALID_ORDER_STATE	ErrorCode = 1280
INVALID_ORIGIN_TYPE	ErrorCode = 1250
INVALID_ORIGINATOR	ErrorCode = 1180
INVALID_POLICIES	ErrorCode = 1531
INVALID_POSITION_EFFECT	ErrorCode = 1240
INVALID_PRICE	ErrorCode = 1160
INVALID_PROCESS_NAME	ErrorCode = 1450
INVALID_PRODUCT	ErrorCode = 1090
INVALID_PRODUCT_TYPE	ErrorCode = 1270
INVALID_QUANTITY	ErrorCode = 1040
INVALID_RATIO_FOR_CROSS_PROD	ErrorCode = 1713
INVALID_REJECT_REQUEST	ErrorCode = 1520
INVALID_SESSION	ErrorCode = 1100
INVALID_SIDE	ErrorCode = 1150
INVALID_SIDE_INDICATOR	ErrorCode = 1152
INVALID_SPREAD	ErrorCode = 1070
INVALID_STATE	ErrorCode = 1110

INVALID_STRATEGY	ErrorCode = 1060
INVALID_STRATEGY_LEG	ErrorCode = 1340
INVALID_THRESHOLD	ErrorCode = 1462
INVALID_TIME	ErrorCode = 1020
INVALID_TIME_BOUNDARIES	ErrorCode = 1535
INVALID_TIME_IN_FORCE	ErrorCode = 1230
INVALID_TRADE_REPORT_HANDLING_INSTRUCTION	ErrorCode = 1464
INVALID_TRADE_SOURCE	ErrorCode = 1410
INVALID_TRADE_TYPE	ErrorCode = 1420
INVALID_TYPE	ErrorCode = 1533
INVALID_UPDATE_ATTEMPT	ErrorCode = 1170
INVALID_UPDATE_PRICE_REPORT	ErrorCode = 1907
INVALID_USER	ErrorCode = 1080
INVALID_USERID_LIST	ErrorCode = 1471
INVALID_USERID_REQUESTING_CANCEL	ErrorCode = 1469
INVALID_USER_ID_FOR_LIGHT_ORDERS	ErrorCode = 1950
INVALID_VERSION	ErrorCode = 1380
INVALID_WORKSTATION_ID	ErrorCode = 1470
LISTENER_ALREADY_REGISTERED	ErrorCode = 1140
MISSING_LISTENER	ErrorCode = 1310
MISSING_SIDE_INDICATOR	ErrorCode = 1151
MULTICLASS_STRATEGY_NOT_ALLOWED	ErrorCode = 1345
NO_REMAINING_QUANTITY	ErrorCode = 1430
NO_TRADE_SO_FAR	ErrorCode = 1802
NO_WORKING_ORDER	ErrorCode = 1135
NOT_AN_ONLY_TRADE	ErrorCode = 1803
NOT_AN_OPENING_ONLY_TRADE	ErrorCode = 1801
ONLY_OPENING_TRADE_SO_FAR	ErrorCode = 1804
ORDER_REJECTED_ON_RSS	ErrorCode = 1909
PREFERENCE_PATH_MISMATCH	ErrorCode = 1120
PRICE_GREATER_THAN_HIGH	ErrorCode = 1809
PRICE_LESS_THAN_LOW	ErrorCode = 1810
PRICE_NOT_EQUAL_TO_LAST_SALE	ErrorCode = 1806
PRICE_NOT_EQUAL_TO_OPENING_PRICE	ErrorCode = 1808

ROOT_ALREADY_EXISTS	ErrorCode = 1472
SIDE_INDICATOR_MISMATCH	ErrorCode = 1153
UNDERLYING_LEG_NOT_LISTED_INSTOCK	ErrorCode = 1712
UNSUPPORTED_ORIGIN_TYPE	ErrorCode = 1711
VOLUME_GREATER_THAN_CUMULATIVE_VOL	ErrorCode = 1811

NotAcceptedCodes

ACTION_VETOED	ErrorCode = 4100
ALREADY_UPDATED_AS_REGISTERED	NotAcceptedCodes = 7007
ALREADY_UPDATED_AS_UNREGISTERED	NotAcceptedCodes = 7008
AUCTION_ENDED	ErrorCode = 4140
AUCTION_INACTIVE	ErrorCode = 4130
DIRECTED_AIM_PRIMARY_EXPIRED	NotAcceptedCodes = 7000
INVALID_AFFILIATED_FIRM	NotAcceptedCodes = 7006
INVALID_REQUEST	ErrorCode = 4020
INVALID_STATE	ErrorCode = 4010
LOCATION_NOT_AVAILABLE	ErrorCode = 4310
MANUAL_QUOTE_ACCEPTED	ErrorCode = 6001
MANUAL_QUOTE_CLASS_NOT_IDX_HYBRID_ENABLED	ErrorCode = 6009
MANUAL_QUOTE_INVALID_REQUEST	ErrorCode = 6006
MANUAL_QUOTE_MARKETABLE	ErrorCode = 6002
MANUAL_QUOTE_MARKETABLE_WITH_STRATEGY	ErrorCode = 6004
MANUAL_QUOTE_NOT_ACCEPTED	ErrorCode = 6007
MANUAL_QUOTE_OVERRIDE_NEEDED	ErrorCode = 6008
MANUAL_QUOTE_SYSTEM_ERROR	ErrorCode = 6005
MANUAL_QUOTE_WORSE_THAN_MARKET	ErrorCode = 6003
NO_AFFILIATED_MM_AVAILABLE_FOR_DIRECTED_AIM	NotAcceptedCodes = 7004
NO_DPM_AVAILABLE_FOR_DIRECTED_AIM	NotAcceptedCodes = 7003
NO_PDPM_AVAILABLE_FOR_DIRECTED_AIM	NotAcceptedCodes = 7002
NOT_REGISTERED_FOR_DIRECTED_AIM	NotAcceptedCodes = 7001
ONLY_USER_FOR_ACRONYM	ErrorCode = 4200
ORDER_BEING_PROCESSED	ErrorCode = 4070
OTHER_USER_FOR_ACR_QUOTING_CLASS	ErrorCode = 4150

PENDING_CANCEL	ErrorCode = 4300
QUOTE_BEING_PROCESSED	ErrorCode = 4060
QUOTE_CONTROL_ID	ErrorCode = 4110
QUOTE_RATE_EXCEEDED	ErrorCode = 4030
RATE_EXCEEDED	ErrorCode = 4040
RECENT_USER_ACTIVITY	ErrorCode = 4240
SEQUENCE_SIZE_EXCEEDED	ErrorCode = 4050
SERVER_NOT_AVAILABLE	ErrorCode = 4080
UNKNOWN_TYPE	ErrorCode = 4000
UNSUPPORTED_INTERNALIZATION	ErrorCode = 4120
USER_HAS_ORDER	ErrorCode = 4230
USER_IS_ENABLED	ErrorCode = 4210
USER_LOGGED_IN	ErrorCode = 4220
USER_NOT_AFFILIATED_TO_ANY_FIRM	NotAcceptedCodes = 7005

NotFoundCodes

RESOURCE_DOESNT_EXIST	ErrorCode = 5000

SystemCodes

PERSISTENCE_FAILURE	ErrorCode = 6000

TransactionFailedCodes

ACTION_VETOED	ErrorCode = 3020
CREATE_FAILED	ErrorCode = 3000
INCOMPLETE_STATE_CHANGE	ErrorCode = 3060
INVALID_STATE_CHANGE	ErrorCode = 3050
UPDATE_FAILED	ErrorCode = 3010