OneChicago, LLC

CBOE direct™ *System Requirements*

Prepared By: Karen Christiansen, Odalys Castro
Version: 0.1
Status: Draft
Save Date: 03/22/02 4:05 PM

1 Table of Contents

2 Introduction

CBOE*direct* will provide trade matching for OneChicago, LLC, the joint venture between the CBOE, CME and CBOT. This joint venture was established as a new exchange to facilitate trading of security futures. Business Requirements for the CBOE systems changes required to support security futures are detailed in CBOE*direct* Functional Requirements. The purpose of this document is to provide the detailed systems requirements to support the Functional Requirements.

This approach leverages the CBOE*direct* functionality and connectivity to equity derivative liquidity providers. Users connected to GLOBEX2 will send orders for Security futures from their current connections. The CME will route such orders to CBOE*direct*, in the CMi 2.0 format, through the pipe pictured above. CBOT users will utilize CBOE*direct* trader workstations. Users connected to CBOE*direct* will send orders and quotes using their current connections.

Order matching will occur on CBOE*direct.* The system will send fill reports back to the originating system. Trade confirmations destined for CME users, as well as market data, will be sent through the pipe in the CMi 2.0 format. CBOE*direct* will send matched trades to the CTM Trade Match system and to CME, where post-trade corrections of non-critical fields may occur. Communication between Trade Match and the CME will be sent through the pipe. The CME's Post-trade processing system will communicate with CBOE via a MQ Series as the primary interface.

CBOE Trade Match will submit trades to OCC and CME for clearance and settlement. CME will build an identical file in the event a back-up is needed. Trades for firms, who are CME members, but not members of the OCC, will be designated for the special clearing account. The current requirements are built upon the assumption that OPRA will disseminate quote and last sale information to the market data vendors.

3 Assumptions

3.1 General Assumptions

3.1.1 ONE Requirements

This document describes functionality required by ONE.

3.1.2 Clearing

3.1.2.1 CME Clearing Systems Changes

The ability of CBOE*direct* users to conduct post-trade processing and/or clear through CME is dependent on completion by CME of the necessary changes to its clearing systems.

3.1.2.2 CTI/Origin

CME will translate CTI/Origin combinations into the CBOE-supported Origins. CBOEdirect will process all quotes for Origin M (Market Maker).

3.1.2.3 Clearing Houses

Both OCC and CME will clear SF transactions for ONE. Transactions of OCC members and their affiliates may be cleared at the OCC. Transactions of CME members, who are not members of OCC, and select transactions of OCC members, will clear in the CME special account at OCC.

3.1.2.4 Matched Trades File

The matched trades file will be sent by CBOE to OCC as soon as possible after 4 p.m.

3.1.2.5 Average Pricing System

Support for an Average Pricing System is not an initial requirement for CBOE*direct*, and if desired by ONE, would be subject to Change Order.

3.1.2.6 Post-Trade Corrections

Firms will make post-trade corrections in the CBOE's CTM system via an ITP correction screen or batch correction. (Post-trade correction in the CME's systems is out of scope of these requirements.) CBOE may provide ONE with master-access to an ITP terminal that would enable ONE to make post-trade corrections on behalf of firms that do not have ITP access but this access is also out of scope of these requirements.

3.1.2.7 CMTA

OCC will provide CMTA facilities to OCC members for SFs. CMTA agreements for SFs will be separate from current equity and index option CMTA agreements.

3.1.2.8 Cross-Clearinghouse Transfers

OCC and the CME Clearing House will support a process to facilitate cross-clearinghouse transfers.

3.1.2.9 Exception Procedures

OCC and CME will provide procedures for post-clearing position movement from/to OCC and CME members for the purposes of error resolution, account transfers and member firm mergers.

3.1.2.10 OCC Dissemination of Reports to CFTC

After CBOE provides the daily matched trade file to OCC, OCC will provide daily reports to the CFTC detailing member firm positions in format acceptable to CFTC.

3.1.2.11 Large Trader Reporting

Development of a customer large trade data report for CFTC reporting purposes is out of scope of this document. OCC will be responsible for providing the OCC compliance tape to the CFTC for Market Maker and Clearing Firm large trade data.

3.1.3 Price Dissemination

The ability of OneChicago to disseminate Security Futures price data is dependent upon completion by the market data processor selected by OneChicago of its development work to support this dissemination. In addition, OneChicago must develop appropriate forms of agreement for data vendors and data subscribers.

ONE market data will be sent by the CBOE to the market data distribution system selected by ONE utilizing a feed from CBOEdirect, through a new adapter, via COPP. However, if the market data distribution system does not utilize the format of the Options Price Reporting Authority (OPRA) or SIAC, the systems changes to support a new format are out of scope.

ONE may choose to send market data through the CME vendor network using the market data supported by the CMi 2.0 interface. CME changes to support price dissemination are out of scope of this document.

3.1.4 Standard Equity Trading Hours

The initial hours of ONE trading will be from 8:30 a.m. until 3:02 p.m. Chicago time.

3.1.5 Back Office Functions

Regulatory/surveillance functions dependent upon receipt of data from CBOE*direct* will be supported by CBOE, but these functions are out of scope of this document.

Billing functions dependent upon receipt of data from CBOE*direct* will be supported by CBOE. These functions are within the scope of this document.

3.1.6 Training

Any participant training requirements, such as user or technical training, are outside the scope of this document.

3.2 Product Assumptions

3.2.1 New Product Entry

ONE will list SFs.

The CBOE Help Desk will add new Contracts, as instructed by ONE, after the close of trading on the previous business day for which the changes are effective.

3.2.2 Decimal Pricing

Pricing of SFs will be in decimals, not fractions, and prices will tick in $0.01 increments.

3.2.3 Price Limits

ONE will not implement daily price limits for SFs. Generally, when a stock is halted, trading in the corresponding SF will be halted as well.

3.2.4 Inter-class Spread Processing Not Supported

Spreads between Contracts of different Product classes will not be supported.

3.3 Access Assumptions

3.3.1 LMM Quote Entry

LMMs will enter quotes via CBOE-provided interfaces, either CMi 2.0 or FIX 4.2. If Market Makers are allowed to provide continuous quotes, they may do so via the GLOBEX Network only if CME supports the quote message type through the CMi 2.0 interface.

LMMs providing continuous quotes will update their bids and offers each time the underlying stock price (last sale) changes.

3.3.2 Market Maker Hand Held Terminals

Market Maker handheld terminals may be utilized to facilitate order entry from the CBOE trading floor for SFs on CBOE*direct*. The process to enter futures orders from such hand held terminals will be similar to what is currently in place to support stock execution on the Chicago Board Options Exchange. Programming efforts that are required to give traders the technology to support such order entry from the trading floor will be the responsibility of the Market Maker clearing firms or stock execution firms.

SF trade correction via market maker hand held terminals will not be supported.

3.3.3 LMM Assignments

ONE will notify CBOE of new LMM assignments before the close on the last business day before the day when the assignments are to be effective. The CBOE membership department will add new LMM assignments to the Membership System after the close of trading on the last business day before the day when ONE requests that the changes will be effective.

3.3.4 CBOT Member Access

CBOT members who do not have access to GLOBEX will access the SF environment via CBOE trader workstations. Interfaces with the a/c/e platform (CBOT's electronic trading platform) or CBOT order routing are not planned for the initial ONE rollout and are out of scope of this document.

3.4 Billing Assumptions

3.4.1 Billing Processing

CBOE will be the main processor for billing of SFs. CBOE's existing "Integrated Billing System", together with appropriate enhancements, will be utilized for this purpose.

Systems work to facilitate billing is within the scope of this document.

3.4.2 Fee Types

ONE will bill for transaction fees. A separate fee structure for non-compliance with LMM obligations, if desired by ONE, would be subject to a Change Order.

No separate fee for trade match is required.

3.4.3 Fee Differentiation

Transaction fees will be differentiated based upon Origin.

Market Makers will be assessed market maker fees for trades in all SFs. No support for product "zones" or "bins" is planned.

3.5 Assumptions Regarding CME Responsibilities

3.5.1 CMi 2.0 Interface

Users connected to the GLOBEX Network may send orders to CBOE*direct* for SFs from their current connections. The CME will route such orders to CBOE*direct*, in the CMi 2.0 format, through the Pipe, and the ability of users to access CBOE*direct* through the GLOBEX Network is dependent on development of the Pipe. Development of the Pipe is out of scope of this document.

3.5.2 Access

ONE requires that users connecting to CBOE*direct* through the GLOBEX Network be able to conduct post-trade allocation and correction in the CBOE's CTM system. To facilitate this, CME must identify GLOBEX users to CBOE*direct* with their CBOE acronym, OCC clearing number and the CBOE exchange identifier.

3.5.3 Non-Supported Clearing Fields

CME will translate CTI/Origin combinations into the OCC-supported Origins before submitting orders to CBOE*direct* and after trade reports are received.

3.5.4 Trade Match Communication

The CME's GLOBEX Trade Processing system ("GTP") will communicate with CBOE via MQ Series as the primary interface.

3.5.5 Billing

ONE Transaction Fee invoices for non-OCC firms will be invoiced to the CME special account at OCC. CBOE will provide billing detail to the CME for further processing.

3.5.6 Support

The CME will provide level one support to CME users accessing ONE through the GLOBEX Network. The GLOBEX Control Center ("GCC") will communicate with the CBOE Help Desk with respect to problems that require CBOE efforts to resolve.

3.5.7 Validation of Give-Up Relationships

CME will maintain a table of valid CMTA to GUS relationships and will validate all allocations for valid relationships. If CBOE sends an invalid allocation to CME, CME will reject the allocation and send a message to CBOE's CTM system.

3.6 Assumptions Regarding OneChicago Responsibilities

3.6.1 Clearing Services Agreements

OneChicago will finalize clearing services agreements with CME and OCC, and OCC and CME will finalize their associate clearinghouse agreement.

3.6.2 OneChicago-CME Pipe Agreement

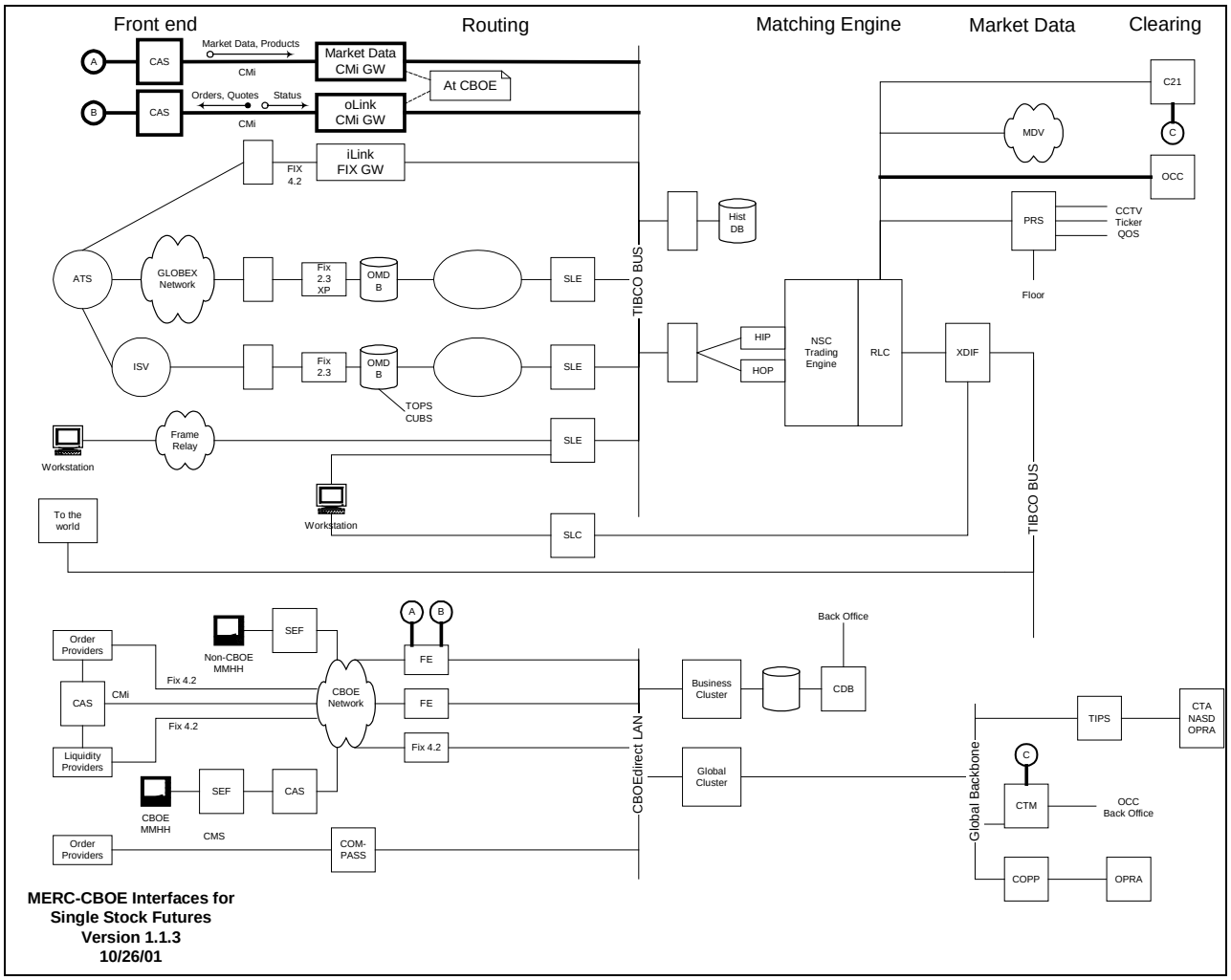
OneChicago will finalize its agreement with CME relating to development and implementation of the Pipe.

3.6.3 Designation of Initial LMMs

OneChicago will designate LMMs for all Products that will be traded on the Launch Date sufficiently in advance of the Launch Date to enable the LMMs to perform necessary systems work.

4 System Flow Diagram

The diagram below depicts the technical environment for connectivity between the CBOE and CME for the trading of security futures.



MERC-CBOE Interfaces for Single Stock Futures Version 1.1.3 10/26/01

5 CBOE*direct* Requirements

5.1 Server Requirements

5.1.1 COMPASS Adapter

The overall strategy is to leverage existing SBT and Compass code as much as possible. To this end, the intent is to use the well-known "HGW Appl To Appl" messaging protocol, which requires the client (CBOE-Direct) to "login" to COMPASS. COMPASS will act as a server / service provider, and CBOE-Direct will act as a client / producer. Both COMPASS and SBT already have code in place to execute the login handshaking sequence, so this code will be re-used. Only the content of the data messages will change.

5.1.1.1 Message Block Framing

A block (collection, multiple msgs, etc.) of transactions may be packaged into a "block frame". The block frame has the following structure:

Field	Length	Data Type	Description
LengthOfBlock	2	short int	The length of the overall block. I think this length should include itself. I.E. if the data after this were 10 bytes of data, the length would be 12. This is the NAPI Message length.
ApplToApplHeader			This is the standard header structure that is widely used throughout CBOE for non-CORBA-based application to application conversations. Particularly between TPF and other applications.
version	1	byte	applToApplHeader version. Set to 0x02 (version two).
frameType	1	byte	applToApplHeader message type. See table of message types below.
originName	8	char	The logical name of the originator of the block. This should be unique across all logical client-side data providers. I.E. if multiple processes / threads / application instances, etc. transmit data to/from COMPASS, each instance will be uniquely identified by having a different name. Primary/Backup process/thread/instances may use the same name, so long as both primary and backup cannot be simultaneously connected.
destName	8	char	This is the "service name" of the COMPASS system. This will probably be the string "ORDER" or "REPORTS" , but should be configurable in case we want to change the name or have different services.
key	4	long	Used by the application to specify the sequence number. This is used for sequence accounting as explained later in this document.
HdrLength	1	byte	The purpose of this field is to "state" the length of the header. Normally this needs to be set to 23 or 0x17 which is the length of the applToApplHeader. If the application chooses, however, it may add additional application-specific header information. that immediately follows this field. If the application does this, then it needs to set this field to a higher value so the hdrLength includes the 23 bytes already here, plus any application header information. For example, if there was a 4 byte application-specific header immediately following this field, the application should set hdrLength to 23 + 4 = 27.

Field	Length	Data Type	Description
			Put the value 0x17 or decimal-23 into this field.
MsgHeader			
LengthOfMsg	5	Char	Length of message in bytes. Includes itself, Required if we block multiple messages. ***(See issues below – Message Blocking)*** Right justified 0 left-filled.
MsgType	1	Char	What kind of message this is, I.E. New Order, Cancel Report, Fill Report etc... See below for values.
Data	N	structure varies	The data for the message goes here. The length and structure of this data will vary depending on the type of transaction which is indicated by the msgType field. Proposed formats for this structure are shown below.
MsgHeader	6		Repeat of msgHeader structure defined above [optional].
Data	M	structure varies	Same as data above. msgHeader.msgType defines which structure the data conforms to, and thus the length [optional].
….			The msgHeader followed by data may repeat up to R times or until the block reaches some maximum length L. R and L should be configurable by CBOE-Direct, or at least R should be configurable so that L may be limited by limiting R. COPP has a maximum buffer length it can process, and at this point I have not researched what the length is. I think it's under 4K, however, but may be in the 3072 range.

5.1.1.2 Frame Types

These are the values for the frameType field described above.

Value Name	Value	Meaning / Usage
CONNECT_PRIMARY	0x01	Client transmits a frame like this to login to compass. This may be used to denote that the client is a primary connection.
CONNECT_SECONDARY	0x02	Client transmits a frame like this to login to COMPASS as a secondary.
CONNECT_ACCEPT	0x03	COMPASS respondes to CONNECT_PRIMARY and CONNECT_SECONDARY with this of the connection is accepted. Name authentication of originName or some severe error on COMPASS will be the basis for whether a connection is accepted or not.
CONNECT_REJECT	0x04	COMPASS transmits this as response to client if it cannot accept the connection at this time.
DISCONNECT_PRIMARY	0x05	Client transmits this to "logout" of COMPASS as a primary client.
DISCONNECT_SECONDARY	0x06	Client transmits this to logout of COMPASS as a secondary (backup) client.
DISCONNECT_ACCEPT	0x07	COMPASS transmits this as a response to either DISCONNECT message above.
DATA	0x08	Block contains Data. Sender does not required acknowledgement. **This will not be used**
DATA_REJECT	0x09	COMPASS/CBOE-Direct will not support this frame type. This is useful in applications where end-to-end data integrity and confirmation is required. This is used to denote an invalid,

Value Name	Value	Meaning / Usage
		unacceptable, or unsupported transaction. ***See Issues below – DATA_REJECTS***
DATA_CONFIRM	0x0A (10)	Block contains data. Sender wants/needs confirmation of receipt of the data. I.E. "Here's some data, please confirm." See ***Issues*** below.
CONFIRM_RESPONSE	0x0B (11)	Prior DATA_CONFIRM frame has been received and sender is no longer responsible for it. (The sequence number in the **key** field will be returned back in this response). See ***Issues*** below.
HEARTBEAT_REQUEST	0x0C (12)	Block contains an "Are you there?" request. Compass can send this (every so often) if it chooses to do heart-beating. The response below will be returned by CBOEDirect.
HEARTBEAT_RESPONSE	0x0D (13)	Block contains an "I am here." response to the HEARTBEAT_REQUEST.
DISCONNECT_REJECT	0x0E (14)	Block contains a reject of a disconnect request. COPP will not support this frame type.

5.1.1.3 Routing Requirements

Compass will route orders over multiple hosts (currently 2 in production). Compass would prefer the reports generated off the order to go over the corresponding host over which the original order was submitted.

Note Exception to the above rule: In the event of a failure of the preferred Compass host, CBOEDirect will route orders to any available host.

5.1.1.3.1 HGW Appl-Appl

There will be 2 types of appl-appl services defined in COMPASS for CboeDirect to use:
ORDERS (Over which Compass will submit orders to CBOEDirect).
REPORTS (Over which CBOEDirect will send reports to Compass).

There can be multiple CBOEDirect clients that will login to Compass over the above mentioned services. Each CBOEDirect client will be identified by a unique name. (This is achieved in the login phase).

Each CBOEDirect client will connect over a separate TCP/IP connection (eg There could be 'n' clients to receive orders from compass and 'm' clients sending reports to compass).

5.1.1.4 Sequence Accounting/Data Validation

Each message sent/received from Compass will contain the sequence number, which is incremented on each message, starting with 1. The sequence number is filled in the **'key'** field in the appl header portion of the message.

Each message sent (from Compass or CboeDirect) will be of the msg type DATA_CONFIRM, the receiving end will respond to the message with a CONFIRM_RESPONSE msg type (sending back the original sequence number submitted in the message) after successful processing of the message.

5.1.1.4.1 Validation Errors

Validation errors will be handled through HAPS as a DATA_REJECT message type. If there is an invalid data error with the order, CBOEdirect will send a copy of the originial order to HAPS. IUs from COMPASS will also be passed through to HAPS. The reject messages will print at the CBOE Help Desk.

5.1.1.4.2 Sequence Number Mismatches

5.1.1.4.2.1 Compass to CBOEDirect

This is the case when Compass is sending orders to CBOEDirect and there is a sequence number mismatch.

Case 1 (Compass Sequence Number Higher then what CBOEDirect is expecting)
In this case the following will occur.

1. CBOEDirect will log an error.
2. CBOEDirect will update its sequence number to match Compass's sequence number.
3. Process the message in the regular fashion.

Case 2 (Compass Sequence Number Lower then what CBOEDirect is expecting)
In this case the following will occur.

1. CBOEDirect will log an error.
2. CBOEDirect will not send a CONFIRM back (Causing Compass to log an error).
3. Manual intervention is required to get out of this condition. *(See issues below on how will we notify support staff that something is wrong).*

5.1.1.4.2.2 CBOEDirect to Compass

This is the case when CBOEDirect is sending reports to Compass and there is a sequence number mismatch.

Case 1 (CBOEDirect Sequence Number Higher then what Compass is expecting)
In this case the following will occur

1. Compass will log an error.
2. Compass will update its sequence number to match CBOEDirect's sequence number.
3. Process the message in the regular fashion.

Case 2 (CBOEDirect Sequence Number Lower then what Compass is expecting)
In this case the following will occur.

1. Compass will log an error.
2. Compass will update its sequence number to match CBOEDirect's sequence number (or Compass does not update the sequence number, but instead sends the message to the firm with the Possible resend or Possible duplicate). *For reports, either case is acceptable since every message is acknowledged. This should not occur for orders since there is no way of identifying duplicate cancel requests in the event of a communication failure).*
3. Process the message in the regular fashion.

5.1.1.5 Data Fields

The various messages required to accept orders and report data have common fields. To avoid duplication, fields will be described once here and these names will be referenced in the **Data Formats** section later in this document.

CompassHeader (for COMPASS internal messages only)

Field Name	Length	Data Type	Meaning / Usage
msgType	2	char	"fe"
mfDevCount	6	char	Number of messages by device.
mfDevId	8	char	MF logical device name
teamTranNum	6	char	Number of message Host LIH
hostDevId	8	char	Host Link device name
busDate	8	char	YYYYMMDD
recvdTime	8	char	Time Read from MF line
sentTime	8	char	time message written to line
servicedFlag	1	char	0, 1, 2, or 4

SBT Header

Field Name	Length	Data Type	Meaning / Usage
MsgHeader			
LengthOfMsg	5	Char	Length of message in bytes, excluding the original input message and length. Includes itself, Required if we block multiple messages. Right justified 0 left-filled.
MsgType	1	Char	What kind of message this is, I.E. New Order, Cancel Report, Fill Report etc... See below for values.
Compass Line Id	8	8 chars	Whatever compass fills in the order will be returned back to compass in this field.
HAPS Printer Id	4	4 chars	The reject destination from the line tin.
Error code	2	2 chars	00 – Success, all others are failures. Defined for posterity since currently we cannot handle rejects.

Fields

Field Name	Length	Data Type	Meaning / Usage
account	16	char	Alpanumeric account number.
branch	3	char	Alpha branch code.
branchSequence	4	char	Numeric branch sequence number.
cmta	4	char	CMTA Number. One alpha for the exchange "O" or "M", followed by 3 numeric OCC or CME account number.
contingency	3	char	Contingency Type number: 001 = None 002 = All or None 003 = Fill or Kill 004 = Immediate or Cancel 010 = Stop Order 012 = On Close 013 = Stop Limit

Field Name	Length	Data Type	Meaning / Usage
contraInfo contraBroker contraFirm contraQuantity	14 3 4 7	char char char char	contraInfo is a structure containing from 1 to 12 of the following: Alpha code for the trade's contra broker. Alpha code for the contra firm mnemonic. Numeric data representing the number of shares for this contra.
correspondentFirm	4	char	Alpha code for original order's correspondent firm.
coverage	1	char	Identifies whether a transaction is covered or uncovered: Covered = "C" Uncovered = "U"
cxlBranch	3	char	Cancel request's Alpha branch code.
cxlBranchSequence	4	char	Cancel request's numeric branch sequence number.
cxlCorrespondentFirm	4	char	Alpha code for cancel request's correspondent firm.
cxlOwnerFirm	4	char	Alpha code for firm canceling the order.
Date Fields orderDate cxlDate	8	char	Numeric date. YYYYMMDD.
errorInputMsg	max = 401	char	The messages consists of the following header: Parsing Errors: " REJ-nnnn-xxxx- l" nnnn = 4 digit offset indicating the number of bytes from the beginning of the error message, including the errorHeader, at which the error occurred. xxxx = device name l = line separator character (0x85). Followed by the original un-processed message received from the firm. Sequence Number Errors: " SEQ.t-xxxx- l" t = 1 character representing the type of sequence number error: H = Skip beyond gap. B = Unreadable sequence number. R = Repeated sequence number. xxxx = device name l = line separator character (0x85). Followed by the original un-processed message received from the firm. If any error occurs during formatting of the input message, the errant message will be forwarded to SBT as type "U".
executionBroker	3	char	Alpha code of the executing broker.

Field Name	Length	Data Type	Meaning / Usage
expirationMonth	3	char	JAN – DEC
expirationYear	4	char	YYYY
filler	20	char	Filler area for future expansion.
InMsgData	max = 401	char	The Original un-processed message received from the firm.
InMsgLength	5	char	Length of the errorInputMsg, or InMsgData.
leavesIndic	1	char	Leaves or Fills indicator. Fills = 'F' Leaves = 'L' Neither = ' '
miscData	32	char	Miscellaneous alphanumeric data included in the original order.
msgHeader	6	char	This is defined above. A msgHeader will preceded each message in a block. It contains the lengthOfMsg and msgType.
numContras	2	char	Numeric data representing the number of contras for this trade.
originator	1	char	Order originating source: Broker/Dealer = "B" Customer = "C" DPM = "D" Firm = "F" CBOE Member = "M"
openClose	1	char	Identifies whether the transaction is Opening or Closing. Opening = "O" Closing = "C" If left blank, "CLOSE" is assumed.
ownerFirm	4	char	Alpha code for firm originating the order.
Price fields orderPrice contingencyPrice originalPrice executionPrice	15	char	All prices will be expressed in a uniform standard. All Prices will be reported in decimals. A price will consist of 7 byte whole + a decimal point + 7 byte fraction. The whole portion will be right-justified left-zero-filled and the fractional portion will be left-justified zero-right-filled.
productTypeCode	1	char	COMMODITY = 1 DEBT = 2 EQUITY = 3 **FUTURE = 4** INDEX = 5 LINKED_NOTE = 6 OPTION = 7 UNIT_INVESTMENT_TRUST = 8 VOLATILITY_INDEX = 9 WARRANT = 10 STRATEGY = 11
session	16	char	Alphanumeric session indicator field.
side	1	char	Buy or Sell Code: Buy = B Sell = S
symbol	8	char	Product symbol / acronym. Left justified, blank-padded.

Field Name	Length	Data Type	Meaning / Usage
Time Fields orderTime executionTime	6	char	Numeric time. HHMMSS
timeInForce	3	char	Time in force indicator: Day Order = DAY Good Until Canceled = GTC Extended Trading = DAA
Volume/Size fields originalQuantity cxlQuantity contingencyQuantity executionQuantity leavesQuantity tlcQuantity	7	char	Right justified left-zero-filled.

5.1.1.6 Data Record Formats

Within a block, multiple messages may appear. Each will be framed by the 2 byte msgHeader (lengthOfMsg + msgType) defined earlier. Messages may be mixed transaction types within the same block.

Below are the recommendations for these structures and the value of the msgType field which defines them.

5.1.1.6.1 Compass To CBOEDirect(Record formats)

Msg Type Value = A (New Order)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	COMPASS	Not be transmitted to SBT.
sbtHeader		20	0	COMPASS	?????
ownerFirm	char	4	20	Compass Line or Order Line 0	executingOrGiveUpFirm ???
correspondentFirm	char	4	24	Order Line 0	correspondentFirm
cmta	char	4	28	Order Line 4B, Field 2/3	cmta
branch	char	3	32	Order Line 1, Field 1	branch
branchSequence	char	4	35	Order Line 1, Field 2	branchSequenceNumber
account	char	16	39	Order Line 4	account
orderDate	char	8	55	Compass system date	orderDate
orderTime	char	6	63	Compass system time	receivedTime
productTypeCode	char	1	69	Order Line 1B, Field 2	ProductType
side	char	1	70	Order Line 2	Side
originalQuantity	char	7	71	Order Line 3, Field 1	originalQuantity
symbol	char	8	78	Order Line 3, Field 2	?????
expirationMonth	char	3	86	Order Line 3, Field 3	expireTime.month

Fields	Type	Size	Offset	Origin	Destination
expirationYear	char	4	89	Order Line 3, Field 3	expireTime.year
orderPrice	char	15	93	Order Line 3, Field 4	price
contingency	char	3	108	Order Line 3, Field 5/7	ContingencyType
contingencyPrice	char	15	121	Order Line 3, Field 6	OrderContingencyStruct.price
congingencyQuantity	char	7	126	"0000000"	
timeInForce	char	3	133	Order Line 3A, Field 1	TimeInForce
originator	char	1	136	Order Line 3B, Field 1	originator
openClose	char	1	137	Order Line 3B, Field 2	OrderState ?????
coverage	char	1	138	Order Line 3B, Field 3	Coverage
session	char	16	139	OrderLine 1B, Field3	?????
miscData	char	32	155	Order Line 4C	optionalData
InMsgLength	char	5	187	Orig. In Msg Len.	Error processing.
InMsgData	char	max= 401	192	Original Input Message.	Error Processing.

Msg Type Value = B (Cancel Order)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	COMPASS	Will not be transmitted to SBT.
sbtHeader		20	0	COMPASS	?????
cxlOwnerFirm	char	4	20	Compass Line or Order Line 0	executingOrGiveUpFirm ???
cxlCorrespondentFirm	char	4	24	Order Line 0	correspondentFirm
cxlBranch	char	3	28	Order Line 1, Field 1	branch
cxlBranchSequence	char	4	31	Order Line 1, Field 2	branchSequenceNumber
cxlDate	char	8	35	Compass system date	?????
cxlQuantity	char	7	43	Order Line 3, Field 1	?????
ownerFirm	char	4	50	Compass Line or Order Line 0	
correspondentFirm	char	4	54	Order Line 0	
branch	char	3	58	Order Line 4A, Field 2	
branchSequence	char	4	61	Order Line 4A, Field 3	
orderDate	char	8	65	Order Line 4A, Field 4	
productTypeCode	char	1	73	Order Line 1B, Field 2	ProductType
side	char	1	74	Order Line 2	Side
symbol	char	8	75	Order Line 3, Field 2	?????
expirationMonth	char	3	83	Order Line 3, Field 3	expireTime.month
expirationYear	char	4	86	Order Line 3, Field 3	expireTime.year

Fields	Type	Size	Offset	Origin	Destination
orderPrice	char	15	90	Order Line 3, Field 4	price
session	char	16	105	OrderLine 1B, Field 3	?????
InMsgLength	char	5	121	Orig. In Msg Len.	Error processing.
InMsgData	char	max = 401	126	Original Input Message.	Error Processing.

Msg Type Value = C (Cancel Replace Order)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	COMPASS	Will not be transmitted to SBT.
sbtHeader		20	0	COMPASS	?????
cxlOwnerFirm	char	4	20	Compass Line or Order Line 0	executingOrGiveUpFirm ???
cxlCorrespondentFirm	char	4	24	Order Line 0	correspondentFirm
cxlBranch	char	3	28	Order Line 1, Field 1	branch
cxlBranchSequence	char	4	31	Order Line 1, Field 2	branchSequenceNumber
cxlDate	char	8	35	Compass system date	?????
cxlQuantity	char	7	43	Order Line 3, Field 1 or Line 3D Field 1	?????
ownerFirm	char	4	50	Compass Line or Order Line 0	
correspondentFirm	char	4	54	Order Line 0	
branch	char	3	58	Order Line 4A, Field 2	
branchSequence	char	4	61	Order Line 4A, Field 3	
orderDate	char	8	65	Order Line 4A, Field 4	
productTypeCode	char	1	73	Order Line 1B, Field 2	ProductType
side	char	1	74	Order Line 2 or Line 3C, Field 2	Side
symbol	char	8	75	Order Line 3, Field 2 or Line 3D, Field 2	?????
expirationMonth	char	3	83	Order Line 3, Field 3 or Line 3D, Field 3	expireTime.month
expirationYear	char	4	86	Order Line 3, Field 3 or Line 3D, Field 3	expireTime.year
orderPrice	char	15	90	Order Line 3C, Field 2 or Line 3D, Field 4	price
session	char	16	105	Order Line 1B, Field 3	
ownerFirm	char	4	121	Compass Line or Order Line 0	executingOrGiveUpFirm ???
correspondentFirm	char	4	125	Order Line 0	correspondentFirm
cmta	char	4	129	Order Line 4B, Field 2/3	cmta
branch	char	3	133	Order Line 1, Field 1	branch
branchSequence	char	4	136	Order Line 1, Field 2	branchSequenceNumber

Fields	Type	Size	Offset	Origin	Destination
account	char	16	140	Order Line 4	account
orderDate	char	8	156	Compass system date	orderDate
orderTime	char	6	164	Compass system time	receivedTime
productTypeCode	char	1	170	Order Line 1B, Field 2	ProductType
side	char	1	171	Order Line 2	Side
originalQuantity	char	7	172	Order Line 3, Field 1	originalQuantity
symbol	char	8	179	Order Line 3, Field 2	?????
expirationMonth	char	3	187	Order Line 3, Field 3	expireTime.month
expirationYear	char	4	190	Order Line 3, Field 3	expireTime.year
orderPrice	char	15	194	Order Line 3, Field 4	price
contingency	char	3	209	Order Line 3, Field 5/7	ContingencyType
contingencyPrice	char	15	212	Order Line 3, Field 6	OrderContingencyStruct.price
contingencyQuantity	char	7	227	"0000000"	
timeInForce	char	3	234	Order Line 3A, Field 1	TimeInForce
originator	char	1	237	Order Line 3B, Field 1	originator
openClose	char	1	238	Order Line 3B, Field 2	OrderState ?????
coverage	char	1	239	Order Line 3B, Field 3	Coverage
session	char	16	240	Order Line 1B, Field 3	?????
miscData	char	32	256	Order Line 4C	optionalData
InMsgLength	char	5	288	Orig. In Msg Len.	Error processing.
InMsgData	char	max = 401	293	Original Input Message.	Error Processing.

Msg Type Value = U (Error: Unprocessable Order)

Fields			Offset	Origin	Destination
compassHeader		55	n/a	COMPASS	Will not be transmitted to SBT.
sbtHeader		20	0	COMPASS	?????
InMsgLength	char	5	20	Orig. In Msg Len.	?????
errorInputMsg	char	max = 401	25	Original Input message that produced the error.	?????

5.1.1.6.2 CBOEDirect To Compass (Record formats)

Msg Type Value = D (Fill Report)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	Not returned from SBT	COMPASS
sbtHeader		20	0	Returned from SBT	COMPASS

Fields	Type	Size	Offset	Origin	Destination
correspondentFirm	char	4	20	correspondentFirm	Report Line 0
cmta	char	4	24	cmta	Report Line 5
branch	char	3	28	branch	Report Line 1, Field 1
branchSequence	char	4	31	branchSequenceNumber	Report Line 1, Field 2
account	char	16	35	account	Report Line 4C
productTypeCode	char	1	51	ProductType	Report Line1A, Field 2
symbol	char	8	52	Symbol	Report Line 3, Field 2
Side	char	1	60	Side	Report Line 2
expirationMonth	char	3	61	expirationDate.month	Report Line 3, Field 3
expirationYear	char	4	64	expirationDate.year	Report Line 3, Field 3
originalPrice	char	15	68	exercisePrice	Report Line 4, Field 2
contingency	char	3	83	OrderContingencyStruct.contingency	Report Line 4, Field 3 or Report Line 4A, Field 2
contingencyPrice	char	15	86	OrderContingencyStruct.price	Report Line 4, Field 4
contingencyQuantity	char	7	101	??????	"0000000"
originator	char	1	108	originator	Report Line 4B, Field 1
openClose	char	1	109	OrderState ?????	Report Line 4B, Field 2
coverage	char	1	110	Coverage	Report Line 4B, Field 3
session	char	16	111		Report Line 1A, Field 3
miscData	char	32	127	optionalData	Report Line 5A
timeInForce	char	3	159	TimeInForce	Report Line 4A, Field 3
executionPrice	char	15	162	FilledReportStruct.price	Report Line 3, Field 4
executionQuantity	char	7	177	tradedQuantity	Report Line 3, Field 1
leavesIndic	char	1	184	?????	Report Line 4A, Field 1
leavesQuantity	char	7	185	leavesQuantity	Report Line 4A, Field 1
executionTime	char	6	192	timeSent	Report Line 6, Field 2
executionBroker	char	3	198	executingBroker	Report Line 6, Field 1
numContras	char	2	201	2 chars from msg	
ContraInfo	char	13	203	1- numContras	Report Line 6, Field 2
filler	char	20	217 – 371		

Msg Type Value = E/F (Cancel Report: UROUT/TLC)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	Not returned from SBT	COMPASS
sbtHeader		20	0	Returned from SBT	COMPASS
correspondentFirm	char	4	20	correspondentFirm	Report Line 0

Fields	Type	Size	Offset	Origin	Destination
cmta	char	4	24	cmta	Report Line 5
branch	char	3	28	branch	Report Line 1, Field 1
branchSequence	char	4	31	branchSequenceNumber	Report Line 1, Field 2
account	char	16	35	account	Report Line 4A
productTypeCode	char	1	51	ProductType	Report Line1A, Field 2
symbol	char	8	52	Symbol	Report Line 3, Field 2
Side	char	1	60	Side	Report Line 2
expirationMonth	char	3	61	expirationDate.month	Report Line 3, Field 3
expirationYear	char	4	64	expirationDate.year	Report Line 3, Field 3
originalPrice	char	15	68	exercisePrice	Report Line 3, Field 4
contingency	char	3	83	OrderContingencyStruct.contingency	Report Line 3, Field 5
contingencyPrice	char	15	86	OrderContingencyStruct.price	Report Line 3, Field 6
contingencyQuantity	char	7	101		"0000000"
originator	char	1	108	originator	Report Line 4, Field 1
openClose	char	1	109	OrderState ?????	Report Line 4, Field 2
coverage	char	1	110	Coverage	Report Line 4, Field 3
session	char	16	111		Report Line 1A, Field 3
miscData	char	32	127	optionalData	Report Line 5A
timeInForce	char	3	159	TimeInForce	Report Line 3B, Field 2
originalQuantity	char	7	162	originalQuantity	Report Line 3, Field 1
cxlQuantity	char	7	169	?????	Report Line 3A, Field 2
tlcQuantity	char	7	184	?????	Used to determine TLC
leavesIndic	char	1	176	?????	Report Line 3B, Field 1
leavesQuantity	char	7	177	?????	Report Line 3B, Field 1
Filler	char	20	191		

Msg Type Value = G (Nothing Done)

Fields	Type	Size	Offset	Origin	Destination
compassHeader		55	n/a	Not returned from SBT	COMPASS
sbtHeader		20	0	Returned from SBT	COMPASS
correspondentFirm	char	4	20	correspondentFirm	Report Line 0
cmta	char	4	24	cmta	Report Line 5
branch	char	3	28	branch	Report Line 1, Field 1
branchSequence	char	4	31	branchSequenceNumber	Report Line 1, Field 2
account	char	16	35	account	Report Line 4A

Fields	Type	Size	Offset	Origin	Destination
productTypeCode	char	1	51	ProductType	Report Line1A, Field 2
symbol	char	8	52	Symbol	Report Line 3, Field 2
side	char	1	60	Side	Report Line 2
expirationMonth	char	3	61	expirationDate.month	Report Line 3, Field 3
expirationYear	char	4	64	expirationDate.year	Report Line 3, Field 3
originalPrice	char	15	68	exercisePrice	Report Line 3, Field 4
contingency	char	3	83	OrderContingencyStruct.contingency	Report Line 3, Field 5
contingencyPrice	char	15	86	OrderContingencyStruct.price	Report Line 3, Field 6
contingencyQuantity	char	7	101		"0000000"
originator	char	1	108	originator	Report Line 4, Field 1
openClose	char	1	109	OrderState ?????	Report Line 4, Field 2
coverage	char	1	110	Coverage	Report Line 4, Field 3
session	char	16	111		Report Line 1A, Field 3
miscData	char	32	127	optionalData	Report Line 5A
originalQuantity	char	7	159	originalQuantity	Report Line 3, Field 1
leavesIndic	char	1	166	?????	Report Line 3B
leavesQuantity	char	7	167	leavesQuantity	Report Line 3B
filler	char	20	174		

5.1.2 Print COMPASS Rejects

COMPASS reject messages will be handled through HAPS as a DATA_REJECT message type. The reject messages will print in real-time at a printer located in the CBOE Help Desk area.

5.1.3 COPP Adapter

5.1.3.1 Stock Futures Reports To COPP/OPRA Design Requirements

5.1.3.1.1 Supported Messages

The following messages will be formatted by the Copp Adapter and sent to OPRA.

- Futures Trade (Over LastSale event channel)
- Futures Quote (Over CurrentMarket channel)
- Futures Summary (Includes Begin/End Summary messages). Trigger is admin request to the adapter. (See below for Summary requirements). The Adapter will query the trading session service for summary values.
- Futures Settlement Value. Trigger is admin request to adapter (See below for settlement value requirements). The Adapter will query the ProductMaintenanceService.
- Text Message (Admin request to adapter)
- End Of Reporting/Good night control message. Trigger is Admin request to the adapter.

5.1.3.1.2 Copp Adapter Configuration

- A Copp Adapter instance should limit the messages it supports to those specified through configuration. This allows processes to be configured to handle subsets of messages in a distributed fashion.
- Configure the queue information. (Queue Name, persistMessage (true or false), maxQueueDepth). If persistMessages is true, create a named persistent queue, else, create a named transient queue.
- Buffering values (maxBufferSize,maxBufferAgeTimeout—Used to flush the buffer to Copp when not full and timeout has expired).
- Names of event channels. If the value is null or not specfied, no connection to the channel will occur. This will allow to further distribute data for performance reasons within a process. (i.e. One connection in the same process can handle Last sale events and the other current market events).

5.1.3.1.3 Admin requests needed/Wrapper Unix scripts

The following admin requests and corresponding wrapper unix scripts are required.

- startSummary sessionName. The Adapter will query TradingSessionService for all classes in the specified session and then for each class (or for a set of classes at a time) query the TradingSessionService again to get the summary information.
- stopSummary sessionName (Abort dissemination immediately)
- sendTextMessage
- sendGoodNight
- startSettlementDissemination sessionName. The Adapter will query TradingSessionService for all classes in the specified session and then for each class (or for a set of classes at a time) query the ProductMaintenanceService to get the settlement information.
- stopSettlementDissemination sessionName (abort dissemination immediately).

5.1.3.2 Futures Summary Requirement

5.1.3.2.1 Data that needs to be sent for summary

(This data will come out of a query to Market Data Service, ProductService and Trading Session Service).

- productSymbol
- sessionId
- expirationMonth
- expirationYear
- priceDenominator
- openInterest
- totalVolume
- openPrice
- highPrice
- lowPrice
- lastPrice
- netChangeInd
- netChange
- bidPrice

- askPrice
- underlyingStockPrice

All the information currently exists in the Market Data service (in the Recap Struct) with the exception of 'underlyingStockPrice' and 'openInterest'.

Note The 'openInterest' field does exists in the RecapStruct , but is never filled in. And will remain that way to avoid cmi changes. We can remove it later when time permits.

To get this data out to OPRA the following services (IDL/code) will need to be modified.

5.1.3.2.2 ProductService

The ProductService will hold onto the 'Open interest' per product. This data will be updated via a file. A new utility will be needed that will update the open interest when operation's executes start of day procedures.

IDL Changes (in product.idl and ProductMaintenanceService.idl)
Interface 'ProductMaintenanceService' add new methods.

```
typedef struct ProductOpenInterestStruct {
        ProductKeysStruct  productKeys;
        long               openInterest;
}
typedef struct ClassOpenInterestStruct {
        ProductOpenInterestStruct[] openInterests;
}
```

a) void updateOpenInterestByProduct(ProductOpenInterestStruct[] openInterests)
b) ProductOpenInterestStruct[] getOpenInterestForProducts(ProductKeyStruct[] productKeys)
c) ClassOpenInterestStruct[] getOpenInterestForClasses(long[] classKey);

5.1.3.2.3 TradingSessionService

The TradingSessionService will act as a controller that accumulates all the summary data and persists it by session.

The futures end of session strategy (which executes at end of session) will do the following for each product in the futures session.

- Query Market Data Service for recap information and corresponding underlying stock price's for each product (by class).
- Query Product Maintenance Service for the open interest.
- Persist the above information.

IDL Changes (TradingSessionService.idl)
interface TradingSessionService add new methods

```
typedef struct ProductSummaryStruct {
        long openInterest;
        RecapStruct recap;
```

}
typedef struct ClassSummaryStruct {
PriceStruct underlyingStockPrice;
ProductSummaryStruct[] summaries;
}
a) ClassSummaryStruct[] getSummaryForClasses(String sessionName, long[] classKeys)

5.1.3.2.4 MarketDataService

For the trading session service to get the recap information with corresponding underlying prices in it, we would like to make the following idl change.

IDL Changes(marketData.idl and MarketDataService.idl)

Interface MarketDataService

a) Change the method RecapStructSequence getRecapForClass(String sessionName, long classKey) to

marketData::ClassRecapStruct getRecapForClass(String sessionName, long classKey)

Where ClassRecapStruct = typedef marketData::ClassRecapStruct{
cmiMarketData::RecapStruct underlyingRecap;
cmiMarketData::RecapStructSequence productRecaps;
}

b) Other code changes for market data service would be
   1) Get the close price from product service for net change calculation (close price is previous days settlement value stored in the product service).

5.1.3.3 Futures Settlement Value Requirements

The following data needs to disseminated to opra for reporting settlement values.
Settlement values will be reported after products close and before session ends.

- productSymbol
- sessionId
- expirationMonth
- expirationYear
- priceDenominator
- tradePrice
- tradeVolume

To get this data out to OPRA the following services (IDL/code) will need to be modified.

5.1.3.3.1 ProductService

The ProductServer will hold onto the 'Settlement prices' per product. This data will be updated either by a gui or via a file. A new utility will be needed that will update the settlement prices before end of Session.

IDL Changes (product.idl and ProductMaintenanceService.idl)

Interface 'ProductMaintenanceService' add new methods.

```
typedef struct ProductSettlementStruct {
        ProductKeysStruct  productKeys;
        PriceStruct        settlementPrice;
}
 typedef struct ClassSettlementStruct {
        ProductSettlementStruct[] settlements;
}
```

a) void updateSettlementByProduct(SettlementStruct[] settlements)
b) SettlementStruct[] getSettlementForProducts(ProductKeysStruct[] productKeys)
c) ClassSettlementStruct[] getSettlementForClasses(long[] classKeys);

5.1.3.3.2 Issues

To get the underlying price we need to setup the underlying product. Currently the underlying product is configured via TIPS product download. The issue is if the underlying products are configured via TIPS download, we will request a TPF options product download. Another way would be **to specify the underlying product** for futures in the file that will be given to us by CME. If using the CME file to get underlying mapping, the current download code will need to be modified so as to not inactivate the underlying products if there are other products that reference the underlying product.

5.1.3.4 Proposed Stock Futures Message Format To COPP

The overall strategy is to leverage existing SBT and COPP code as much as possible. The intent is to use the well-known "HGW Appl To Appl" messaging protocol, which requires the client (CBOE-Direct) to "login" to COPP. COPP will act as a server / service provider, and CBOE-Direct will act as a client / producer. Both COPP and SBT already have code in place to execute the login handshaking sequence, so this code will be re-used. Only the content of the data messages will change.

5.1.3.4.1 Message Block Framing

A block (collection, multiple msgs, etc.) of transactions may be packaged into a "block frame". The block frame has the following structure:

Field	Length	Data Type	Description
LengthOfBlock	2	short int	The length of the overall block. I think this length should include itself. I.E. if the data after this were 10 bytes of data, the length would be 12.
ApplToApplHeader			This is the standard header structure that is widely used throughout CBOE for non-CORBA-based application to application conversations. Particularly between TPF and other applications.
version	1	byte	applToApplHeader version. Set to 0x02 (version two).
frameType	1	byte	applToApplHeader message type. See table of message types below.
originName	8	char	The logical name of the originator of the block. This should be unique across all logical client-side data providers. I.E. if multiple processes / threads / application instances, etc. transmit data to COPP, each instance should be uniquely identified by having a different name. Primary/Backup

Field	Length	Data Type	Description
			process/thread/instances may use the same name, so long as both primary and backup cannot be simultaneously connected.
destName	8	char	This is the "service name" of the COPP system. This will probably be the string "COPP", but should be configurable in case we want to change the name or have different services.
key	4	long	May be used by the application for any way it sees fit. A possible use might be for CBOE-Direct to put the order-number or some type of sequence number into this field which could be useful in diagnosing issues when trades or quotes are claimed to have not been reported, etc. This would give us a way to match things up between COPP and CBOE-Direct. This field could go unused, however.
hdrLength	1	byte	The purpose of this field is to "state" the length of the header. Normally this needs to be set to 23 or 0x17 which is the length of the applToApplHeader. If the application chooses, however, it may add additional application-specific header information. that immediately follows this field. If the application does this, then it needs to set this field to a higher value so the hdrLength includes the 23 bytes already here, plus any application header information. For example, if there was a 4 byte application-specific header immediately following this field, the application should set hdrLength to 23 + 4 = 27. It is this author's opinion that this field has been and is being mis-used because in order to locate this field you must offset 23 bytes into the message in the first place, and thus you know there are 23 bytes here anyway. So having to put 23 in this field is pretty useless if you have no additional application header data after this field. Unfortunately this is the way the field has been used and both COPP and SBT code assume the 23 will be there. It is used to offset in memory from the beginning of the header to locate the application data. Put the value 0x17 or decimal-23 into this field.
msgHeader			
lengthOfMsg	1	byte	Length of message in bytes. Includes itself.
msgType	1	byte	What kind of message this is, I.E. quote, trade, summary, settlement, etc. See below for values.
data	N	structure varies	The data for the message goes here. The length and structure of this data will vary depending on the type of transaction which is indicated by the msgType field. Proposed formats for this structure are shown below.
msgHeader	2	2 bytes	Repeat of msgHeader structure defined above [optional].
data	M	structure varies	Same as data above. msgHeader.msgType defines which structure the data conforms to, and thus the length [optional].
….			The msgHeader followed by data may repeat up to R times or until the block reaches some maximum length L. R and L should be configurable by CBOE-Direct, or at least R should be configurable so that L may be limited by limiting R. COPP has a maximum buffer length it can process, and at this point I have not researched what the length is. I think it's under 4K, however, but may be in the 3072 range.

5.1.3.4.2 Frame Types

These are the values for the frameType field described above:

Value Name	Value	Meaning / Usage
CONNECT_PRIMARY	0x01	Client transmits a frame like this to login to COPP. This may be used to denote that the client is a primary connection.
CONNECT_SECONDARY	0x02	Client transmits a frame like this to login to COPP as a secondary.
CONNECT_ACCEPT	0x03	COPP respondes to CONNECT_PRIMARY and CONNECT_SECONDARY with this of the connection is accepted. Name authentication of originName or some severe error on COPP will be the basis for whether a connection is accepted or not.
CONNECT_REJECT	0x04	COPP transmits this as response to client if it cannot accept the connection at this time.
DISCONNECT_PRIMARY	0x05	Client transmits this to "logout" of COPP as a primary client.
DISCONNECT_SECONDARY	0x06	Client transmits this to logout of COPP as a secondary (backup) client.
DISCONNECT_ACCEPT	0x07	COPP transmits this as a response to either DISCONNECT message above.
DATA	0x08	Block contains Data. Sender does not required acknowledgement.
DATA_REJECT	0x09	COPP will not support this frame type. This is useful in applications where end-to-end data integrity and confirmation is required. This is used to denote an invalid, unacceptable, or unsupported transaction.
DATA_CONFIRM	0x0A (10)	Block contains data. Sender wants/needs confirmation of receipt of the data. I.E. "Here's some data, please confirm." Open issue: This is useful in applications where end-to-end confirmation is required, and I don't think we have this case here. We need to discuss this.
CONFIRM_RESPONSE	0x0B (11)	Prior DATA_CONFIRM frame has been received and sender is no longer responsible for it. Open issue: This is useful in applications where end-to-end confirmation is required.
HEARTBEAT_REQUEST	0x0C (12)	Block contains an "Are you there?" request.
HEARTBEAT_RESPONSE	0x0D (13)	Block contains an "I am here." response to the HEARTBEAT_REQUEST. I am open to suggestion on whether this is necessary or not.
DISCONNECT_REJECT	0x0E (14)	Block contains a reject of a disconnect request. COPP will not support this frame type.

5.1.3.4.3 Fields

The various messages required to report quotes, trades, and summary data have common fields. To avoid duplication, fields will be described once here and these names will be referenced in the **Data Formats** section later in this document.

Field Name	Length	Data Type	Meaning / Usage
msgHeader	2	2 chars	This is defined above. A msgHeader will preceded each message in a block. It contains the lengthOfMsg and msgType.
productSymbol	8	char	Product symbol / acronym. Left justified, blank-padded.
sessionId	1	char	Session identified. Blank = "regular" or day session. Other values may be used if early-trading-hours or after-hours sessions

Field Name	Length	Data Type	Meaning / Usage
			are envisioned.
expirationMonth	1	byte	Ordinal number of expiration month. 0x01 through 0x0B.
expirationYear	1	byte	A binary byte representing the last digit column of the year. 0x00 through 0x09
quoteCondition	1	char	A character representing the conditions under which a quote was generated. See OPRA specification for the list of possible conditions. CBOE-Direct may encode this any way they choose. I.E. ordinal internal representation of conditions, re-mapped to character conditions, etc. COPP will translate these into OPRA specification format via a table. COPP needs to know the possible values for this table, which can be changed for future needs. Valid Values are T – Trade Halt F – Fast Market Space – No condition, regular.
tradeCondition	4	char	A character string representing the conditions for the trade. This is also referred to as the "prefix". These will be mapped by COPP via a table lookup. In the event a lookup fails, COPP should report an error, but format the message as a "regular" trade and send it to OPRA anyway.
priceDenominator	1	byte	A byte which says how many decimal places the product trades-in. All prices in any message will be reported with the same denominator code. I.E. if the product trades in minimum price values of $0.05, the number of decimal places is 2. 0x00 means whole-dollars, 0x01 means $1/10^{th}$ dollars, 0x02 means $1/100^{th}$ dollars (pennies), etc. up to a maximum of 0x07, which is all the OPRA specification allows for. A value of 0x20 (ASCII space) will be interpreted to mean that COPP will supply a default denominator code to OPRA when reporting the price. COPP will have a configurable parameter (initially set to pennies) to use. Increment 1 of CBOE-Direct Futures may not have time to obtain this information, so COPP will be coded to provide a default. This way, we can function with minimal effort.
tradePrice	14	See ***Prices*** below	The price of the trade.
tradeVolume	6	See ***Sizes*** below	The volume of a trade.
bidPrice	14	See ***Prices*** below	The bid price of a quote.
askPrice	14	See ***Prices*** below	The ask price of a quote.
bidSize	6	See ***Sizes*** below	The size of the bid.
askSize	6	See ***Sizes*** below	The size of the ask.
openInterest	6	See ***Sizes*** below	The open interest for this futures product.
totalVolume	6	See ***Sizes*** below	The total volume traded for this futures product.
highPrice	14	See ***Prices***	The highest price of the day for the product.

Field Name	Length	Data Type	Meaning / Usage
		below	
lowPrice	14	See ***Prices*** below	The lowest price of the day for the product.
lastPrice or closePrice	14	See ***Prices*** below	The final sale price of the day, I.E. the closing price. **Note:** For futures this may actually be the settlement price.
openPrice	14	See ***Prices*** below	The opening trade price for the day.
netChange	14	See ***Prices*** below	The different between the lastPrice / closePrice and the lastPrice / closePrice from the prior trading day.
netChangeInd	1	char or byte	The sign (direction) of the netChange. Values: + (up), - (down) or 0 (unchanged).
underlyingStockPrice	14	See ***Prices*** below	The underlying stock price.

5.1.3.4.4 Sizes

All sizes will be a uniform standard data type and length. The data type will be a char[6], right-justified zero left-filled. Values that exceed 999,999 should be sent as 999,999.

5.1.3.4.5 Prices

All prices will be expressed in a uniform standard. A price will consist of 2 char[7] fields. A whole portion and a fractional portion. No explicit decimal point will exist in the price. The whole portion will be right-justified left-zero-filled and the fractional portion will be left-justified zero-right-filled.

5.1.3.5 Data Record Formats

Within a block, multiple messages may appear. Each will be framed by the 2 byte msgHeader (lengthOfMsg + msgType) defined earlier. Messages may be mixed transaction types within the same block. At a minimum, the transaction types that must be supported suggest a minimum of four "structures", each having slightly different content. Below are the recommendations for these structures and the value of the msgType field which defines them.

5.1.3.5.1 Msg Type Value 0x01 Futures Trade

Structure:

- msgHeader
- productSymbol
- sessionId
- expirationMonth
- expirationYear
- priceDenominator
- tradeCondition
- tradePrice
- tradeVolume

5.1.3.5.2 Msg Type Value 0x02 Futures Quote

Structure:

- msgHeader
- productSymbol
- sessionId
- expirationMonth

- expirationYear
- priceDenominator
- quoteCondition
- bidPrice
- bidSize
- askPrice
- askSize

5.1.3.5.3 Msg Type Value 0x03 Futures Summary

Structure:

- msgHeader
- productSymbol
- sessionId
- expirationMonth
- expirationYear
- priceDenominator
- openInterest
- totalVolume
- openPrice
- highPrice
- lowPrice
- lastPrice
- netChangeInd
- netChange
- bidPrice
- askPrice
- underlyingStockPrice

5.1.3.5.4 Msg Type Value 0x04 Futures Settlement Value

Note This has the same structure as the Futures Trade message to reduce the number of distinct structures. The volume field should be zero-filled (however zero is represented for size fields). COPP will ignore the tradeVolume for these message types.

Structure:

- msgHeader
- productSymbol
- sessionId
- expirationMonth
- expirationYear
- priceDenominator
- tradeCondition
- tradePrice
- tradeVolume

5.1.3.5.5 Msg Type Value 0x05 Begin Summary Control Message

Structure:

- msgHeader

No data after the lengthOfMsg + msgType header is needed. COPP will format the entire message.

5.1.3.5.6 Msg Type Value 0x06 End Summary Control Message

Structure:

- MsgHeader

No data after the lengthOfMsg + msgType header is needed. COPP will format the entire message.

5.1.3.5.7 Msg Type Value 0x07 Text Message

Structure:

- msgHeader
- ASCII Text

This message type is optional. Since we were defining specifications between CBOE-Direct and COPP, we might as well define and allow for this. It may not be coded by either side.

5.1.3.5.8 Msg Type Value 0x08 End Of Reporting / "Good Night" Control Message

Structure:

- msgHeader

Note No data after the lengthOfMsg + msgType header is needed. COPP will format the entire message.

5.1.4 Product Maintenance

CBOEdirect will support the maintenance of product. The system will allow new products to be defined and edited. To support this functionality, the Product Maintenance Service will be modified to include the following methods.

- addProducts
- addProductClass
- addReportingClass
- updateProductType
- updatePriceAdjustment
- getPriceAdjustment

5.1.5 Stop/Stop Limit Support

CBOEdirect will support Stop and Stop Limit order types for Security Futures. The next two sections detail the system process for Stop and Stop Limit orders.

5.1.5.1 Stop Orders

Stop orders will be processed by the system as follows:

- The order is entered into the system.
- The server validates the order as a Stop (STP) order.
- The server processes the order as a 'New Order' and maintains the state of the order as inactive until the product for the order trades at the stop price or better.
- When the limit price is reached, the server matches the order against the highest priority order(s) from the opposite side of the book at an equal or better price. This converts the order to a market order.
- The CBOEdirect user interface will display the order information in the "My Best" column of the market display only to the trader who submitted the order.

5.1.5.2 Stop Limit Orders

Stop Limit orders will be processed by the system as follows:

- The order is entered into the system.
- The server recognizes the order as a stop limit (STP LIMIT) order.
- The server processes the order as a 'New Order' and maintains the state of the order as inactive until the product for the order trades at the stop price or better.
- When the limit price is reached, the server matches the order against the highest priority order(s) from the opposite side of the book at an equal or better price. This converts the order to a limit order.
- The CBOEdirect user interface will display the order information in the "My Best" column of the market display only to the trader who submitted the order.

5.1.6 Require Sub Account for Order Entry

CBOEdirect requires the customer account number at the time of order entry. This information will be entered in the sub-account field. The system will validate sub-account information as follows:

- An order entry request is submitted.
- The server validates the customer account number in the sub-account field.
- The server proceeds with order entry processing.
- If the sub-account information is invalid because it is missing or because there are too many characters in the field, the server will send an error message to the user interface.

5.1.7 Support for Multiple Exchanges

The existing database structure will be modified to support unique users and firms. The table below provides detailed information for each of the database tables.

Table Name	Column Name	Data Type	Comments
product	open_interest	number(10)	Used for futures trading.
	settlement_price	varchar2(12)	Used for futures trading.
	prod_sub_type_code	number(2)	Used for strategy products; valid values defined in Table 1 of the Appendix.
exchange	membershipKey	number(20)	Unique key to the membership exchange table.
	product_type_number		Deleted
	is_home_number		Deleted
	exchange_name	varchar2(50)	Changed Data Type from *varchar2(32)* to *varchar2(50)*
	acronym	varchar2(5)	Changed Data Type from varchar2(8) to varchar2(5)
firm	exchangeAcronym	varchar2(5)	Exchange acronym for firm.
	firmNumber	varchar2(5)	Changed data type from varchar2(3) to varchar2(5)
sbt_user	userId	varchar2(15)	Intended to replace *user_name* field for naming consistency. The *user_name* field is retained for compatibility.
	exchange_acronym	varchar2(5)	Changed data type from varchar2(8) to varchar2(5)
trading_property	session_key	number(20)	Associate trading property to session instead of exchange.
	exchange_key		Deleted
trading_session	exchange_acronym	varchar2(5)	
userHistory	userId	varchar2(15)	Changed Data Type from *varchar2(12)* to *varchar2(15)* for consistency.

Table Name	Column Name	Data Type	Comments
userSession	userId	varchar2(15)	Changed Data Type from *varchar2(20)* to *varchar2(15)* for consistency.
best_quote	bid_exch	varchar2(5)	Changed Data Type from *varchar2(8)* to *varchar2(5)* for consistency.
	ask_exch	varchar2(5)	Changed Data Type from *varchar2(8)* to *varchar2(5)* for consistency.
cur_mkt	exchange	varchar2(5)	Changed Data Type from *varchar2(8)* to *varchar2(5)* for consistency
order_book_price_deta	quote_user_id	varchar2(15)	Changed Data Type from *varchar2(3)* to *varchar2(15)* for consistency
rfq_history	userId	varchar2(15)	Changed Data Type from *varchar2(4)* to *varchar2(15)* for consistency
sbt_quote	userId	varchar2(15)	Intended to replace the *memberKey* field which holds the user acronym. The *memberKey* field is retained for compatibility.
	clearing_firm_key		The *clearing_firm_key* field should be removed. It is retained for compatibility.
sbtquotehistory	userId	varchar2(15)	Intended to replace the *memberKey* field which holds the user acronym. The *memberKey* field is retained for compatibility.
	event_type		**See details in section 5.1.7.2**
sbttradehistory			Changes still pending. Open issue: determine if any changes will be required since the table is deprecated.
sbt_user_report	userFirmKey	number(20)	Intended to replace *firmKey*. The *firmKey* field is retained for compatibility.
	executingOrGiveupFirmKey	number(20)	Intended to replace *executingOrGiveupFirm*. The field *executingOrGiveupFirm* is retained for compatibility.
	userId	varchar(15)	Changed Data Type from *varchar2(20)* to *varchar2(15)* for consistency.
sbt_user_report_ack	useId	varchar2(15)	Intended to replace *userName* for naming consistency. The field *userName* is retained for compatibility.
sbt_user_report_user	useId	varchar2(15)	Intended to replace *userName* for naming consistency. The field *userName* is retained for compatibility.
sbtorder	executingOrGiveupFirmKey	number(20)	Intended to replace *executingOrGiveupFirm*. The field *executingOrGiveupFirm* is retained for compatibility.
	CmtaExchAcr	varchar2(5)	Required to support futures trading.
	originatorExchAcr	varchar2(5)	Required to support futures trading.
	userId	varchar2(15)	Changed Data Type from *varchar2(5)* to *varchar2(15)* for consistency.
	originator	varchar2(15)	Changed Data Type from *varchar2(5)* to *varchar2(15)* for consistency.
	crossedExecutingOrGiveupFirm		Deprecated. Retained for compatibility.
	crossedBranch		Deprecated. Retained for compatibility.
	crossedBranchSequenceNumber		Deprecated. Retained for compatibility.
	crossedCorrespondentFirm		Deprecated. Retained for compatibility.
	crossedOrderDate		Deprecated. Retained for compatibility.
sbtOrderHistory	cmtaExchAcr	varchar2(5)	Required to support futures trading.
	userId	varchar2(15)	Changed Data Type from *varchar2(4)* to *varchar2(15)* for consistency.

Table Name	Column Name	Data Type	Comments
	event_type		**See details in section 5.1.7.1**
sbt_tradereportentry	active	char(1) not null	Indicates whether this trade report entry is active.
	entry_type	char(1)	A=Add; D= Delete.
	entry_time	number(24)	Time entry was created
	last_entry_type	char(1)	Set to *entry_type* of new trade report entry if this entry is being replaced (i.e. as part of a bust).
	last_update_time	number(24)	Last time any trade report entry with this matched sequence number was modified
	buy_broker_exch	varchar2(5)	Exchange code for the broker acronym in the *buyer* field.
	sell_broker_exch	varchar2(5)	Exchange code for the broker acronym in the *seller* field.
	buy_firm_exch	varchar2(5)	Exchange code for the firm number in the *buyFirm* field.
	buy_firm_branch	varchar2(5)	Branch code from retail order.
	buy_firm_branch_seq_no	number(10) not null	Branch sequence number from retail order.
	buy_cmta_exch	varchar2(5)	Exchange code for the firm number in *buyer_cmta*.
	buy_acct	varchar2(16)	Buyer account.
	buy_corr_id	varchar2(5)	Alpha; Buyer correspondent firm acronym.
	buy_originator	varchar2(5)	Buyer originator broker acronym.
	buy_originator_exch	varchar2(5)	Exchange code for the acronym in *buy_originator*.
	sell_firm_exch	varchar2(5)	Exchange code for the seller firm number in the *sellFirm* field.
	sell_firm_branch	varchar2(5)	Branch code from retail order.
	sell_firm_branch_seq_no	number(10) not null	Branch sequence number from retail order.
	sell_cmta_exch	varchar2(5)	Exchange code for the firm number in *seller_cmta*.
	sell_acct	varchar2(16)	Seller account.
	sell_corr_id	varchar2(5)	Alpha; Seller correspondent firm acronym.
	sell_originator	varchar2(5)	Seller originator broker acronym.
	sell_originator_exch	varchar2(5)	Exchange code for the acronym in *sell_originator*.
	buyer	varchar2(10)	Changed Data Type from *varchar2(20)* to *varchar2(10)* for consistency
	seller	varchar2(10)	Changed Data Type from *varchar2(20)* to *varchar2(10)* for consistency
sbt_traderreport	trade_type	char(1)	B=Block Trade; E=Exchange for Physical; Space=Regular Trade.
mkt_data_summary	databaseIdentifier	number(20)	Unique identifier for row.
	underlying_price	varchar2(12)	Underlying price.
	open_interest	number(10)	The open interest on this product.
	session_name	varchar2(30)	The session for which this summary is relevant.
	prod_key	number(20)	The unique identifier of the product.
	prod_type_code	number(2)	The product type.
	recap	(number(20)	Foreign key to the recap information.
SBTOrderLegDetail	databaseIdentifier	number(20)	Unique identifier for row.
	productKey	number(20)	Key of product for strategy leg.
	orderDBId	number(20)	Key of strategy order – will have foreign key with cascade on delete.
	side	char(1)	Side for this leg; B=Buy; S=Sell.
	originalQuantity	number(10)	Strategy order original quantity * leg ratio.
	tradedQuantity	number(10)	Total traded for this leg.

Table Name	Column Name	Data Type	Comments
	cancelledQuantity	number(10)	Total cancelled for this leg.
	clearingFirmKey	number(20)	Key of clearing firm to be used if leg is a stock.
	mustUsePrice	varchar2(20)	Contingency price for leg. Not supported in SBT; will be sent to TPF.
	positionEffect	char(1)	Open/Close code for leg; O=Open; C=Closed; N=Not Applicable
	coverage	char(1)	Covered/Uncovered code for leg (only required for sell legs); C=Covered; U=Uncovered; B=Unspecified
trade_report_ack	databaseIdentifier	number(20) not null	Unique identifier for row.
	atomic_trade_id	number(20) not null	Row of sbt_tradereportentry.
	matched_seqno	number(20) not null	Matched sequence number.
	entry_type	char(1)	Entry type of sbt_tradereportentry.
	ack_ind	char(1)	Boolean indicating whether report was accepted.
	error_flags	number(20) not null	Integer representation of error flags.

5.1.7.1 New values for event_type in sbtorderhistory

Below are the new event_type values that are supported in the sbtorderhistory table.

Constant Name	Value	Description
NEW_ORDER_STRATEGY_LEG	51	History entry for leg of new strategy order.
FILL_STRATEGY_LEG	52	Trade information for leg of a strategy order.
CANCEL_STRATEGY_LEG	53	Cancel information for leg of a strategy order.
BUST_STRATEGY_LEG_FILL	54	Information for leg of strategy order that was part of a busted trade.
BUST_REINSTATE_STRATEGY_LEG	55	Information for leg of strategy order that was reinstated in the bust processing for a trade.
UPDATE_STRATEGY_LEG	57	New values for a leg of a strategy order that was updated.
PRICE_ADJUST_ORDER_LEG	60	New values for a leg of a strategy order that was affected by a price adjustment.

5.1.7.2 New values for event_type in sbtquotehistory

Below are the new event_type values that are supported in the sbtquotehistory table.

Constant Name	Value	Description
QUOTE_LEG_FILL	152	Trade information for the leg of a quote on a strategy product.
BUST_QUOTE_LEG_FILL	157	Information for leg of a quote on a strategy product that was part of a busted trade.

5.1.7.3 Values for Exchange Codes in CBOEdirect

CBOEdirect will support an "Exchange Identifier" in order to maintain uniqueness among ONE member acronyms and firms. The table below lists the Exchange Identifiers supported in CBOEdirect.

Code	Meaning
CBOE	Chicago Board Options Exchange
CBOT	Chicago Board of Trade
AMEX	American Stock Exchange
BSE	Boston Stock Exchange
CSE	Cincinnati Stock Exchange
CHX	Chicago Stock Exchange
NASD	National Association of Securities Dealers
NYSE	New York Stock Exchange
NYME	New York Mercantile Exchange
LIFFE	International Financial Futures and Options Exchange
PHLX	Philadelphia Stock Exchange
PSE	Pacific Stock Exchange
CME	Chicago Mercantile Exchange
ISE	International Stock Exchange

5.1.8 CTM Futures Database Interface

The CTM data specifications for acknowledge records and input fields for futures trading is detailed below.

5.1.8.1 SBT - CTM Acknowledgement Record

FIELD NAME	SBT X-REF	BYTES	FMT	DESCRIPTION
SBT Trade Server Routing Group	**SBT Trade Server Routing Group**	10	N	For SBT internal routing
SBT Trade Report ID	SBT Trade Report ID	20	N	For SBT use
Transaction Code	Transaction Code	1	X	A = add (original record) D = delete (bust)
Matched Sequence Number	Matched Sequence Number	7	N	Unique Atomic ID
Process OK Switch	Process OK Flag	1	X	O = OK; N = Not OK
Unique ID Switch	Invalid Matched Sequence Number	1	X	* = not OK
Delete Request Not OK Switch	Invalid Delete Request	1	X	* = Matched Seq. Nbr. not found
Duplicate Record Switch	**Duplicate Record**	1	X	* = Duplicate Matched Sequence Nbr.
Transaction Code Not OK Switch	Invalid Transaction Code	1	X	* = not OK
Series Not OK Switch	Invalid Series	1	X	* = not OK
Trade Price Not OK Switch+	Invalid Premium (?)	1	X	* = not OK
Trade Date Not OK Switch	Invalid Trade Date	1	X	* = not OK
Trade Quantity Not OK Switch	Invalid Trade Quantity	1	X	* = not OK
Buy Clearing Mbr Nbr Not OK Switch	Invalid Buyer Firm	1	X	* = not OK
Buy Account Type Not OK Switch	Invalid Buyer Account Type	1	X	* = not OK
Buy Sub-Account Not OK Switch	Invalid Buyer Account	1	X	* = not OK
Buy CMTA Not OK Switch	Invalid Buyer CMTA	1	X	* = not OK
Buy Open-Close Not OK Switch	Invalid Buyer Open/Close Indicator	1	X	* = not OK
Buy Executing Broker Not OK Switch	Invalid Buyer Executing Broker	1	X	* = not OK
Sell Clearing Mbr Nbr Not OK	Invalid Seller Firm	1	X	* = not OK

FIELD NAME	SBT X-REF	BYTES	FMT	DESCRIPTION
Switch				
Sell Account Type Not OK Switch	Invalid Seller Account Type	1	X	* = not OK
Sell Sub-Account Not OK Switch	Invalid Seller Account	1	X	* = not OK
Sell CMTA Not OK Switch	Invalid Seller CMTA	1	X	* = not OK
Sell Open-Close Not OK Switch	Invalid Seller Open/Close Indicator	1	X	* = not OK
Sell Executing Broker Not OK Switch	Invalid Seller Executing Broker	1	X	* = not OK
Business Date Not OK Switch		1	X	* = not OK
Buy Clearing Mbr Exch Not OK Switch		1	X	* = not OK
Buy CMTA Exchange Not OK Switch		1	X	* = not OK
Sell Clearing Mbr Exch Not OK Switch		1	X	* = not OK
Sell CMTA Exchange Not OK Switch		1	X	* = not OK
Filler		21	X	

Record length = 75 bytes

This record layout will be used for Options at some point in the future.
+ **Trade Price field is used for Premium Price in Options and for Futures Price in Futures**

5.1.8.2 SBT - CTM Data Record

FIELD NAME	SBT X-REF	BYTES	FMT	DESCRIPTION
Version		1	X	Distinguish new layout versions
SBT Trade Server Routing Group	SBT Trade Server Routing Group**	10	N	For SBT internal routing
SBT Trade Report ID	SBT Trade Report ID**	20	N	For SBT use
Transaction Code	Transaction Code	1	X	A = add D = delete (bust)
Matched Sequence Number	Matched Sequence Number	7	N	Unique Atomic ID
SBT Session ID	SBT Session ID	10	X	
Trading Symbol	Trading Symbol	12	X	Initially 6 characters max
Expiration Date	Expiration Date	8	N	CCYYMMDD
Strike Price Dollar*	Strike Price Dollar	6	N	
Strike Price Fraction*	Strike Price Fraction	4	X	
Put/Call Code*	Call/Put Indicator	1	X	
Trade Price Dollar+	Trade Price Dollar	6	N	
Trade Price Decimal+	Trade Price Fraction	4	X	
Execution Trade Date	Execution Trade Date	8	N	CCYYMMDD
Execution Trade Time	Execution Trade Time	8	N	HHMMSSff
Business Trade Date	Business Trade Date	8	N	
Contract Quantity	Trade Quantity	7	N	
Market Multiplier	Market Multiplier (Contract Size)	7	N	
Product Line Code	Product Line Code	2	X	
Security Type Code	Security Type Code	2	X	

FIELD NAME	SBT X-REF	BYTES	FMT	DESCRIPTION
Special Trade Indicator	Special Trade Indicator	1	X	B = block trade [space] E = exchange for physical
Filler	Filler	19	X	
Buy Firm	Buyer Firm	5	N	
Buy Firm Exchange	Buyer Firm Exchange	5	X	CBOE; CME; CBOT
Buy Account Type	Buyer Account Type	1	X	C = customer other codes F = firm may be possible M = market maker B = broker dealer
Buy Q/Joint Account	Buyer Account	10	X	Q-account or joint account
Buy Subaccount	Buyer Subaccount	10	X	MM account / customer account
Buy CMTA Firm	Buyer CMTA Firm	5	N	Zero if no CMTA
Buy CMTA Firm Exchange	Buyer CMTA Firm Exchange	5	X	CBOE; CME; CBOT
Buy Open/Close Indicator	Buyer Open/Close Indicator	1	X	O = open C = closed space = undetermined
Buy Broker Acronym	Buyer Acronym	5	X	
Buy Broker Acronym Exchange	Buy Broker Acronym Exchange	5	X	
Buy Trade Originator	Buyer Originator	5	X	Initiator of buy trade
Buy Trade Originator Exchange	Buyer Originator Exchange	5	X	
Buy Firm Branch	Buyer Firm Branch	5	X	
Buy Firm Branch Sequence Number	Buyer Firm Branch Sequence Nbr	10	X	
Buy Optional Data	Buyer Optional Data	32	X	
Buy Correspondent ID	Buyer Correspondent ID	5	X	May be a CMTA acronym
Filler	Filler	10	X	
Sell Firm	Seller Firm	5	N	
Sell Firm Exchange	Seller Firm Exchange	5	X	CBOE; CME; CBOT
Sell Account Type	Seller Account Type	1	X	C = customer other codes F = firm may be possible M = market maker B = broker dealer
Sell Q/Joint Account	Seller Account	10	X	Q-account or joint account
Sell Subaccount	Seller Subaccount	10	X	MM account / customer account
Sell CMTA Firm	Seller CMTA Firm	5	N	Zero if no CMTA
Sell CMTA Exchange	Seller CMTA Firm Exchange	5	X	CBOE; CME; CBOT
Sell Open/Close Indicator	Seller Open/Close Indicator	1	X	O = open C = closed

FIELD NAME	SBT X-REF	BYTES	FMT	DESCRIPTION
				space = undetermined
Sell Broker Acronym	Seller Acronym	5	X	
Sell Broker Acronym Exchange	Sell Broker Acronym Exchange	5	X	
Sell Trade Originator	Seller Originator	5	X	Initiator of sell trade
Sell Trade Originator Exchange	Seller Originator Exchange	5	X	
Sell Firm Branch	Seller Firm Branch	5	X	
Sell Firm Branch Sequence Number	Seller Firm Branch Sequence Nbr	10	X	
Sell Optional Data	Seller Optional Data	32	X	
Sell Correspondent ID	Seller Correspondent ID	5	X	May be CMTA acronym
Filler	Filler	10	X	

Record length = 400 bytes
This record layout will be used for Options at some point in the future.
*** These fields only used for Options**
**** Use is only for reference within SBT**
+ Trade Price fields are used for Premium Price in Options and for Futures Price in Futures

5.1.9 Dynamic Book Depth Update

CBOEdirect will support dynamic book depth updates up to n(5) price levels deep. Updates to the book depth will occur when there is a market best change within the system. Events that will cause a market best change include:

- Cancel Order
- Cancel Replace Order
- New Limit Order
- Fill Order
- Cancel Quote

The server will publish the new top of the book information to the event channel.

5.1.10 User Enablement

CBOEdirect will restrict the level of user functionality by session or by product. A set enablement feature will be designed for this function. The new database will be setup as follows:

Name	Null?	Type
DATABASEIDENTIFIER	NOT NULL	NUMBER(20)
USER_ID	NOT NULL	VARCHAR2(15)
USER_KEY		NUMBER(20)
SESSION_NAME		VARCHAR2(30)
PRODUCT_TYPE		NUMBER(2)

5.1.11 Block Trading and EFPs

Block trading and EFPs will be supported in CBOEdirect. The following system changes will be required:

- Create an acceptTrades method
- Create a query struct to pass to the CAS
- Add a start time/end time method to limit the amount of information forwarded to the CAS
- Define block trades and EFP validation codes (see below)

5.1.11.1 Validation Codes

Field Name	Req.	Detail	Order Validation	EFP/BT Validation	Sample Data
Session ID	Yes		OrderValidationStrategy should exist for the Session		ONE_MAIN
Contract ID (includes Product and Expiration Month)	Yes	CAS will do a lookup based on Contract ID for ProductKey	Our ProductKey is an int. Validated by TradingProductHome		
Price	Yes		Validated for a Product, and Type of Order		
Quantity	Yes		Always +ve, 0 allowed with flag set.	Always +ve If Block Trade Indicator indicates BT, Quantity >= N (Configurable)	
EFP/Block Trade Indicator	Yes	tradeType		TradeType != Regular Trade	
Execution Time	Yes	User Entered or defaults to current time			
Trade Date	Yes	System Assigned		Trade Date = Current Business Date	
Buy Executing Firm (or Give up Firm)	Yes	Must be a CBOE/CME/CBOT member Firm	If MM – check against Profile. executingGiveupFirm Else – check against userStruct.executingGiveupFirm		
Buy Executing Firm Exchange Indicator	Yes			Limted to CBOE, CME, CBOT	CBOE, CME, CBOT
Buy Broker Acronym	Yes			3 Chars per MembershipAdapterImpl?	These 2 fields will be used to do a lookup against the UserId table.
Buy Broker Exchange ID	Yes			CBOE, CME, CBOT	
Buy Origin	Yes	order.orderOriginType	Should be uppercase	Validate against cmiConstants.OrderOrigins	V, E, Q, F, G, H, R, C, O, T, M, B, X
Buy Customer Account Number (Sub-Account)	Yes		Field Length limited by DB, SUBACCOUNT_MAX_LENGTH	Max 10 Characters	
Buy Account	No		If MM – check against Profile.account Else – Field Length limited by DB, ACCOUNT_MAX_LENGTH	Max 16 Characters	

Field Name	Req.	Detail	Order Validation	EFP/BT Validation	Sample Data
Buy CMTA Firm	No	order.cmta, available only if Clearing Firm is different from Executing Firm	Field Length limited by DB, CMTA_MAX_LENGTH		
Buy CMTA Exchange ID	No		Field Length limited by DB, CMTA_MAX_LENGTH	CBOE, CME, CBOT	CBOE, CME, CBOT
Buy Open/Close Indicator	No	order.positionEffect	OPEN, CLOSED, NOTAPPLICABLE		
Buy (Firm) Correspondent ID	No	When a Broker enters order on behalf of another Firm orderId.correspondent Firm	Should be upto 4 uppercase characters		
Buy Optional Data	No		Not greater than OPTIONAL_DATA_MAX_LENGTH	Not greater than 32 (max supported by CTM)	
Buy Originator	No	When a Broker enters order on behalf of another Broker order.originator	Through UserService		
Sell Executing Firm	Yes	Must be a CBOE/CME/CBOT member Firm	If MM – check against Profile. executingGiveupFirm Else – check against userStruct.executingGiveupFirm		
Sell Executing Firm Exchange ID	Yes			Limted to CBOE, CME, CBOT	CBOE, CME, CBOT
Sell Broker Acronym	Yes			3 Chars per MembershipAdapterImpl?	These 2 fields will be used to do a lookup against the UserId table.
Sell Broker Exchange ID	Yes			CBOE, CME, CBOT	
Sell Customer Account Number (Sub-Account)	Yes		Field Length limited by DB, SUBACCOUNT_MAX_LENGTH	Max 10 Characters	
Account	No		If MM – check against Profile.account Else – Field Length limited by DB, ACCOUNT_MAX_LENGTH	Max 16 Characters	
Sell Origin	Yes	order.orderOriginType	Should be uppercase	Validate against cmiConstants.OrderOrigins	V, E, Q, F, G, H, R, C, O, T, M, B, X
Sell CMTA Firm	No	order.cmta, available only if Clearing Firm is different from Executing Firm	Field Length limited by DB, CMTA_MAX_LENGTH		
Sell CMTA Firm Exchange ID	No		Field Length limited by DB, CMTA_MAX_LENGTH	CBOE, CME, CBOT	CBOE, CME, CBOT

Field Name	Req.	Detail	Order Validation	EFP/BT Validation	Sample Data
Sell Open/Close Indicator	No	order.positionEffect	OPEN, CLOSED, NOTAPPLICABLE		
Sell Correspondent ID	No	When a Broker enters order on behalf of another Firm orderId.correspondentFirm	Should be upto 4 uppercase characters		
Sell Optional Data	No		Not greater than OPTIONAL_DATA_MAX_LENGTH	Not greater than 32 (max supported by CTM)	
Sell Originator	No	When a Broker enters order on behalf of another Broker order.originator	Through UserService		

5.1.12 Calculate Settlement Price

Settlement price calculations will be performed at the end of trading for each Contract having open interest. CBOEdirect will provide the following information for each contract:

- Product Symbol
- Month ID
- Year ID
- Settlement Price

The rules initially established by ONE for calculating the settlement price are as follows. If ONE wishes to modify the settlement price algorithm, such modification will be subject to a Change Order.

5.1.12.1.1 Initial Algorithm for Daily Settlement Price Calculation for Day in Which Trades Occur

If the last trade price is equal to or in-between the final bid and offer of the day, the last trade price will be used to populate the settlement price.

If the last trade price is lower than the final bid price, the final bid price will be used to populate the settlement price.

If the last trade price is higher than the final offer price, the final offer price will be used to populate the settlement price.

5.1.12.1.2 Initial Algorithm for Daily Settlement Price Calculation for Day in Which No Trades Occur

If there is no last trade price, an alternate approach must be used to calculate settlement price. In such a case, the previous day's settlement price will be used in lieu of the last trade price to determine the current day's settlement price.

If the previous day's settlement price is lower than the final bid price, the final bid price will be used to populate the settlement price.

If the previous day's settlement price is higher than the final offer price, the final offer price will be used to populate the settlement price.

5.1.12.2 Daily Settlement Price Calculation Under Special Circumstances

If trading is halted for one or more Contracts, but trading in the underlying securities continues, or if trading in any security underlying a particular Contract is halted after an opening market price has already been disseminated for that Contract, the daily settlement price will be determined in accordance with the Rules, subject to OCC's rules.

5.1.12.3 Final Settlement Price Calculation

The final settlement value will be the price of the underlying stock at the close of trading of the SFs or at a configurable time determined by ONE.

If trading is halted for a Contract, on the day of final settlement, and trading in the underlying security on the primary exchange continues, the final settlement value will be the price of the underlying stock at the close of trading of the Contract on ONE or at a time determined by ONE.

If trading is halted in the underlying stock, on the day of final settlement, the settlement price will be the next available opening price of the underlying stock. Settlement will be delayed until the opening of the underlying stock and will be manually entered into the system by an ONE designee. If it is determined that the stock will not resume trading, ONE will determine the final settlement price. In this case, the settlement price will be manually entered into CBOEdirect.

These provisions with respect to final settlement price are subject to OCC's rules.

5.1.13 Disseminate Settlement Price to CME and OCC

5.1.13.1 Record Layout

Record Position	Note	Field Name/Description	Size	Value/ Format	Decimal
01 - 02		Record Id	02	'10'	
03 - 10		Processing Date	08	N	YYYYMMDD
11 - 12		Exchange Code	02	N	Submitting Exchange 07 = ONE
13 - 13		Product Type Code	01	X	'X' = Options on Future 'F' = Future 'F' = Security Future
14 - 15		Filler Space	02	X	
16 - 16		Call/ Put Code	01	'C', 'P'	~~b~~ for Futures
17 - 18		Filler Space	02	X	
19 - 24		Future Trading Symbol	06	X	I
25 - 26		Filler Space	02	X	
27 - 28		Expiration Month	02	MM	
29 - 30		Filler Space	02	X	
31 - 34		Expiration Year	04	CCYY	
35 - 36		Filler Space	02	X	
37 - 41		Strike Price Integer	05	N	~~b~~ for Futures
42 - 45		Strike Price Decimal	04	X	~~b~~ for Futures
46 - 55		Filler Space	10	X	
56 - 64		Mid- Day Price	09	N	4
65 - 68		Filler Space	04	X	

Record Position	Note	Field Name/Description	Size	Value/ Format	Decimal
69 – 77		Settlement Price	09	N	4
78 – 81		Filler Space	04	X	
82 – 82		Settle On Open Indicator	01	X	' * ' or ~~b~~
83 - 200		Filler Space	118	X	

5.1.14 Send Series Add File to OCC

5.1.14.1 Header Record Layout

Record Position	Note	Field Name/ Description	Size	Format	Value
01 – 01		Record ID	01	PIC X	H = Header
02 – 03		Batch Number	02	PIC 9	01 – 99
04 – 11		Batch Date	08	PIC 9	YYYYMMDD
12 – 15		Exchange ID	04	PIC X	Submitting Exchange 'ONE'
16 – 18		Transaction ID	03	PIC 9	531
19 – 99		Filler Space	81	PIC X	
100 – 100	1	Edit Status	01	PIC X	
101 – 200		Filler Space	100	PIC X	

Notes:

[1] Reserved for OCC use

5.1.14.2 Detail Record Layout

Record Position	Note	Field Name/ Description	Size	Format	Value
01 – 01	1, 2	Record ID	01	PIC X	D = Detail
02 – 03	1, 2	Batch Number	02	PIC 9	01 – 99
04 – 11	1, 2	Batch Date	08	PIC 9	YYYYMMDD
12 – 15	1, 2	Exchange ID	04	PIC X	Submitting Exchange 'ONE'
16 – 19	1, 2	Sequential Detail Number	04	PIC 9	0001 – 9999
20 – 22	1,2	Transaction ID	03	PIC 9	531
23 –23	1, 2	Transaction Code	01	PIC X	A=Add C=Change D=Delete P=Position Limit Change
24 – 29	1, 2	Class Symbol	06	PIC X	
30 – 30	1, 2	Class Type Code	01	PIC X	'O' = Option 'X' = Option on Future 'F' = Future
31 – 31		Class Type Unique ID	01	PIC X	
32 – 35	2	Contract Expiration Year	04	PIC 9	YYYY
36 – 37	2	Contract Expiration Month	02	PIC 9	MM
38 – 39		Filler Space	02	PIC X	

Record Position	Note	Field Name/ Description	Size	Format	Value
40 – 44		Contract Strike Price Integer	05	PIC 9	
45 – 46		Contract Strike Price Fraction	02	PIC 9	
47 – 48		Call Ticker Symbol	02	PIC X	
49 – 52		Filler Space	04	PIC X	
53 – 54		Put Ticker Symbol	02	PIC X	
55 – 58		Filler Space	04	PIC X	
59 – 66	2	Contract Activate Date	08	PIC 9	
79 – 79		Filler Space	01	PIC X	
80 – 91	1	Near Term Position Limit	12	PIC 9	
92 – 99		Filler Space	08	PIC X	
100 – 100	3	Edit Status	01	PIC X	
101 – 200		Filler Space	100	PIC X	

Notes:

[1] These represent the only fields that are required for a transaction code of "P" which indicates a position limit change.
[2] These represent the only fields that are required for a class type code of "F" which indicates the product is futures.
[3]Reserved for OCC use.

5.1.14.3 Trailer Record Layout

Record Position	Note	Field Name/ Description	Size	Format	Value
01 – 01		Record ID	01	PIC X	T = Trailer
02 – 03		Batch Number	02	PIC 9	01 – 99
04 – 11		Batch Date	08	PIC 9	YYYYMMDD
12 – 15		Exchange ID	04	PIC X	Submitting Exchange 'ONE'
16 – 19		Hash Total Record Count	04	PIC 9	
20 – 25		Hash Total Expiration Years	06	PIC 9	
26 – 31		Hash Total Expiration Months	06	PIC 9	
32 – 41		Hash Total Strike Integers	10	PIC 9	
42 – 51		Hash Total Strike Fractions	10	PIC 9	
52 – 99		Filler Space	48	PIC X	
100 – 100	1	Edit Status	01	PIC X	
101- 200		Filler Space	100	PIC X	

Note:
[1] Reserved for OCC use.

5.1.15 Expand List of Supported Origins

CME-specific origin codes will be supported by CBOEdirect. The table below details the proposed origin combinations.

CTI/Origin Combination	Proposed CBOE-Origin	Proposed OCC-Origin
CTI 1/Origin 1	M	M
CTI 1/Origin 2	E	M
CTI 1/Origin 5	Q	M
CTI 2/Origin 2	F	F
CTI 3/Origin 1	G	M
CTI 3/Origin 2	H	M
CTI 3/Origin 5	R	M
CTI 4/Origin 1	C	C
CTI 4/Origin 2	O	C
CTI 4/Origin 5	T	C

5.1.16 Capture and Report Performance Statistics

5.2 CMi Requirements

5.2.1 Require Sub Account for Order Entry

CMi users will be required to enter the customer account number at the time of order entry. This information will be entered in the sub-account field. The system will validate sub-account information during order entry processing.

5.2.2 Dynamic Book Depth Update

CMi users will be able to retrieve dynamic book depth information by subscribing to the current market by product key and session name. Subscription by class will not be supported. CBOEdirect will support dynamic book depth updates up to n (5) price levels deep.

5.3 CAS Requirements

5.3.1 Require Sub Account for Order Entry

The CAS will receive sub-account validation from the server. Sub-account information will be cached with order entry processing.

5.3.2 Dynamic Book Depth Update

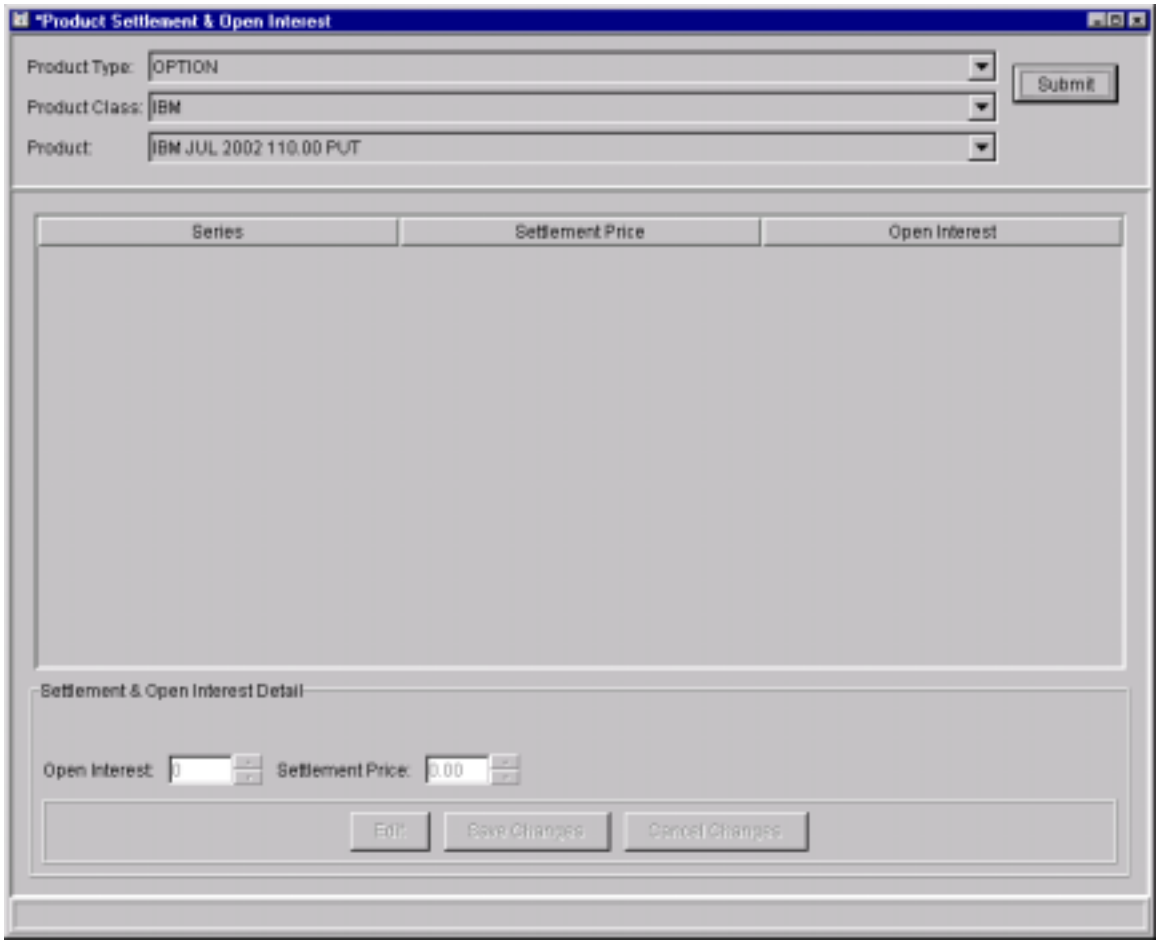
The CAS will cache the updated book depth information it receives from the server. Book depth information will be published by product key and session name. Publication by class will not be supported. CBOEdirect will support dynamic book depth updates up to n (5) price levels deep.

5.4 SA GUI Requirements

5.4.1 Settlement Price

Daily settlement price calculations will be performed at the end of trading for each contract having open interest. The user interface below displays the settlement price and open interest

information by product type, product class and product. Settlement price and open interest modifications can be made in the 'Settlement and Open Interest Detail' section of the screen.



5.4.2 Product Maintenance

A CBOEDirect SAGUI user will be able to create new Product Classes and Products, as well as modify existing product information. The convention for displaying ONE single stock futures on the CBOE direct GUI will be:

- Reporting Class: 3-character
- Expiration month abbreviation
- Expiration Year

Example: March 2002 IBM futures will look like: IBM1C MAR 2002

The following two sections detail the single stock Futures that will be listed by ONE Chicago and narrow based indexes.

5.4.2.1 Listing of Single Stock Futures

Stock Name	Product Class	Reporting Class	Expiration Cycle		
			Jan	Feb	Mar

Stock Name	Product Class	Reporting Class	Expiration Cycle		
			Jan	Feb	Mar
American Express	AXP	AXP1C	X		
American International Group	AIG	AIG1C		X	
Amgen Inc.	AMGN	AMGN1C	X		
AMR Corp/Del	AMR	AMR1C		X	
AOL Time Warner, Inc.	AOL	AOL1C	X		
Applied Materials	AMAT	AMAT1C	X		
AT & T Corporation	T	T1C	X		
Bank of America	BAC	BAC1C		X	
Bank One	ONE	ONE1C		X	
Best Buy Company Inc.	BBY	BBY1C			X
Biogen Inc.	BGEN	BGEN1C	X		
Bristol-Myers Squibb Co.	BMY	BMY1C			X
Broadcom Corp-Cl A	BRCM	BRCM1C		X	
Brocade Communications Systems	BRCD	BRCD1C	X		
Cephalon Inc.	CEPH	CEPH1C		X	
Check Point Software Tech.	CHKP	CHKP1C	X		
Chevron Texaco Corp.	CVX	CVX1C			X
Cisco Systems, Inc.	CSCO	CSCO1C	X		
Citigroup, Inc.	C	C1C			X
Coca-Cola Company	KO	KO1C		X	
Dell Computer Corporation	DELL	DELL1C		X	
eBay, Inc.	EBAY	EBAY1C	X		
EMC Corporation	EMC	EMC1C	X		
Emulex Corp.	EMLX	EMLX1C	X		
Exxon Mobil Corporation	XOM	XOM1C	X		
Ford Motor Company	F	F1C			X
General Electric Company	GE	GE1C			X
General Motors Corp.	GM	GM1C			X
Genzyme Corp – Genl Division	GENZ	GENZ1C	X		
Goldman Sachs Group, Inc.	GS	GS1C	X		
Halliburton Co.	HAL	HAL1C	X		
Home Depot Inc.	HD	HD1C		X	
Idec Pharmaceuticals Corp.	IDPH	IDPH1C	X		
Intel Corporation	INTC	INTC1C	X		
International Business Machines Corporation	IBM	IBM1C	X		
InVision Technologies Inc.	INVN	INVN1C	X		
J.P. Morgan Chase & Co.	JPM	JPM1C			X
Johnson & Johnson	JNJ	JNJ1C	X		
KLA-Tencor Corporation	KLAC	KLAC1C			X
Krispy Kreme Doughnuts Inc.	KKD	KKD1C		X	
Merck & Co., Inc.	MRK	MRK1C	X		
Merrill Lynch & Co., Inc.	MER	MER1C	X		
Micron Technology Inc.	MU	MU1C	X		
Microsoft Corporation	MSFT	MSFT1C	X		
Morgan Stanley Dean Witter	MWD	MWD1C	X		
Motorola, Inc.	MOT	MOT1C	X		
Newmont Mining Corp. Hldg. Co.	NEM	NEM1C			X

Stock Name	Product Class	Reporting Class	Expiration Cycle		
			Jan	Feb	Mar
Nokia Corporation ADR	NOK	NOK1C	X		
Northrop Grumman Corp.	NOC	NOC1C		X	
Novellus Systems Inc.	NVLS	NVLS1C			X
Oracle Corporation	ORCL	ORCL1C			X
PepsiCo Inc.	PEP	PEP1C	X		
Pfizer	PFE	PFE1C			X
Philip Morris	MO	MO1C			X
Proctor & Gamble Co.	PG	PG1C	X		
Qlogic Corp.	QLGC	QLGC1C	X		
Qualcomm, Inc.	QCOM	QCOM1C	X		
SBC Communications Inc.	SBC	SBC1C	X		
Schlumberger Ltd.	SLB	SLB1C		X	
Siebel Systems, Inc.	SEBL	SEBL1C		X	
Sprint Corp-PCS Group	PCS	PCS1C		X	
Starbucks Corp.	SBUX	SBUX1C	X		
Sun Microsystems	SUNW	SUNW1C	X		
Symantec Corp.	SYMC	SYMC1C	X		
Texas Instruments Incorporated	TXN	TXN1C	X		
Tyco International Ltd.	TYC	TYC1C	X		
UAL Corp.	UAL	UAL1C		X	
VERITAS Software Corporation	VRTS	VRTS1C		X	
Verizon Communications Inc.	VZ	VZ1C	X		
Wal-Mart Stores Inc.	WMT	WMT1C			X
Xilinx Inc.	XLNX	XLNX1C			X

5.4.2.2 Narrow Based Indexes

Index Name	Index Components	Product Class	Reporting Class	Expiration Cycle		
				Jan	Feb	Mar
Airlines	AMR					
	CAL					
	DAL					
	LUV					
	UAL					
Investment Banking	GS					
	LEH					
	MER					
	MWD					
Biotech	AMGN					
	BGEN					
	CHIR					
	GENZ					
	HGSI					
Computers	AAPL					
	DELL					
	IBM					
	RIMM					
	SUNW					
Oil Services	BHI					

				Expiration Cycle		
	BJS					
	HAL					
	SLB					
	WFT					
Retail	AZO					
	BBY					
	CC					
	HD					
	WMT					
Defense	GD					
	LMT					
	NOC					
	RTN					
Semiconductor Components	BRCM					
	INTC					
	MU					
	TXN					
	XLNX					

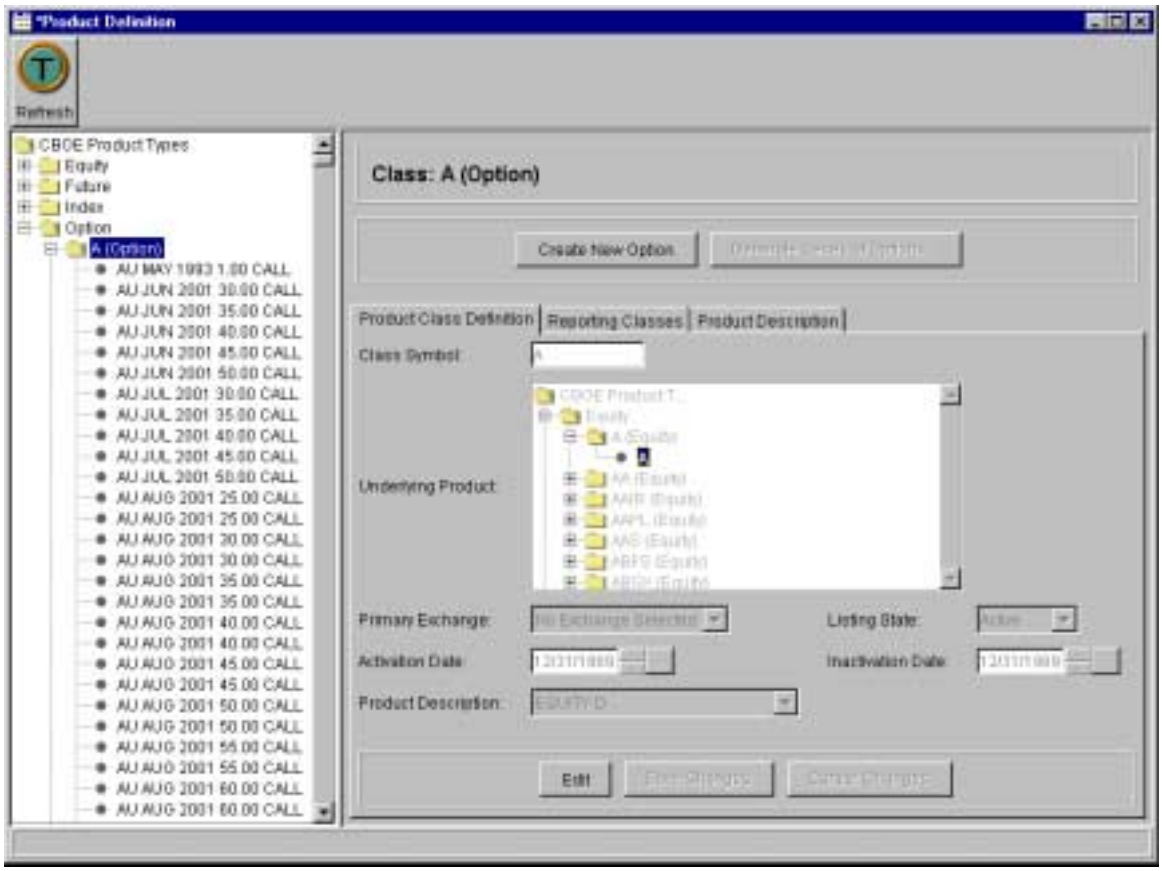
5.4.2.3 Define Product Class

Below are examples of the user interface and the necessary steps to define/modify product information.

5.4.2.3.1 Edit Product Class Definition

The screen below portrays the input fields required to edit Product Classes.

- Select the Product Type (Equity, Future, Option) of the new Product Class.
- The user will then enter information for three categories: 'Product Class Definition', 'Reporting Classes', and 'Product Description'. The required information for each of the categories may vary depending on the Product Type.



- Define the Product Class
  - Information for an Equity Class
  - Class Symbol
  - Primary Exchange
  - Listing State
  - Activation Date
  - Inactivation Date
  - Product Description
- Information for a Future or Option Class
  - Class Symbol
  - Underlying Product
  - Primary Exchange
  - Listing State
  - Activation Date
  - Inactivation Date
  - Product Description

5.4.2.3.1.1 Fields

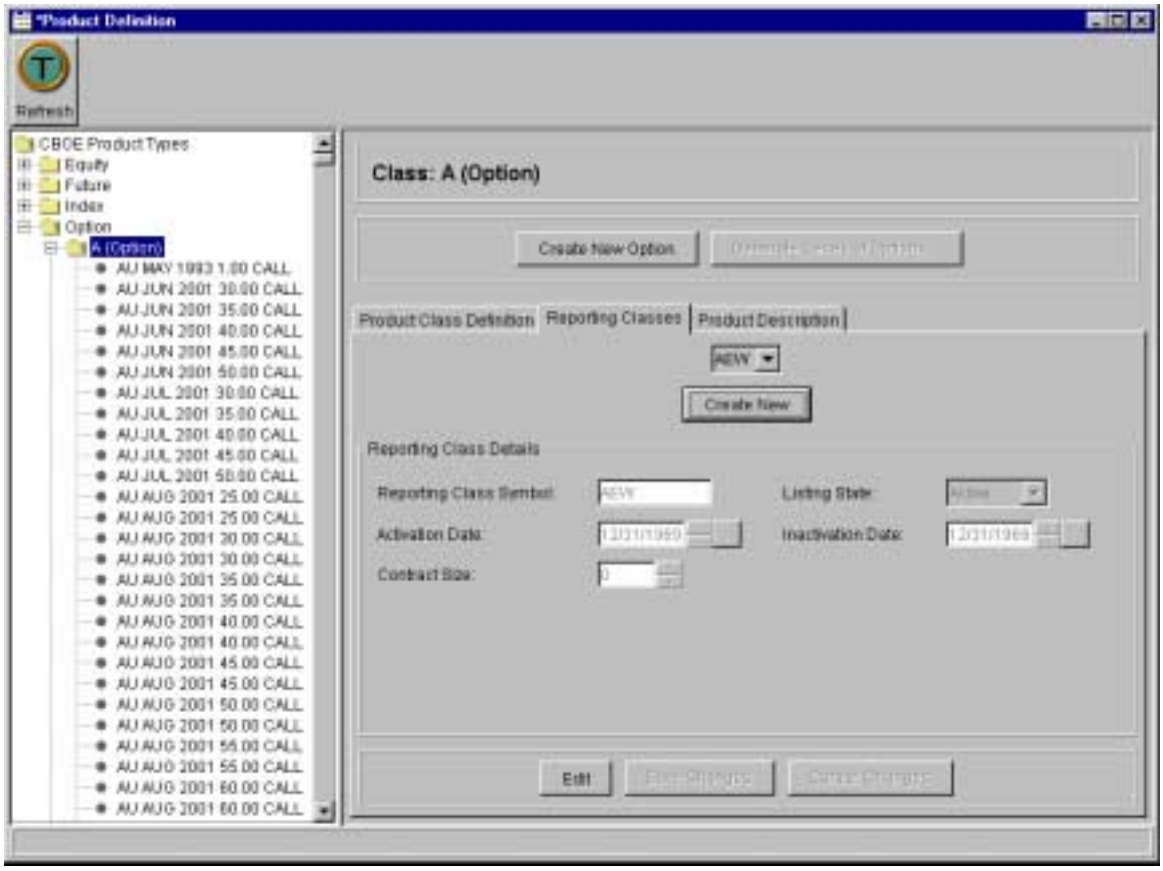
The table below lists the fields and input field types required for each type of product.

Field	Input Type	Equity	Future	Index	Option
Class Symbol	text	x	x	x	x
Underlying Product	product tree		x		x
Primary	drop-down	x	x	x	x

Field	Input Type	Equity	Future	Index	Option
Exchange					
Listing State	drop-down	x	x	x	x
Activation Date	date spinner	x	x	x	x
Inactivation Date	date spinner	x	x	x	x
Product Description	drop-down	x	x	x	x

5.4.2.3.2 Edit Reporting Classes

This user interface depicts the input fields required to edit Reporting Classes.



- Define Reporting Classes
  - A default Reporting Class will have been created by the system when the Product Class was created. This user can choose to edit this default. For *Options* Product Classes only, the user can also create additional Reporting Classes. Other product types can have only one Reporting Class.
  - Information for a Reporting Class
    - Reporting Class Symbol
    - Listing State
    - Activation Date
    - Inactivation Date

- Contract Size

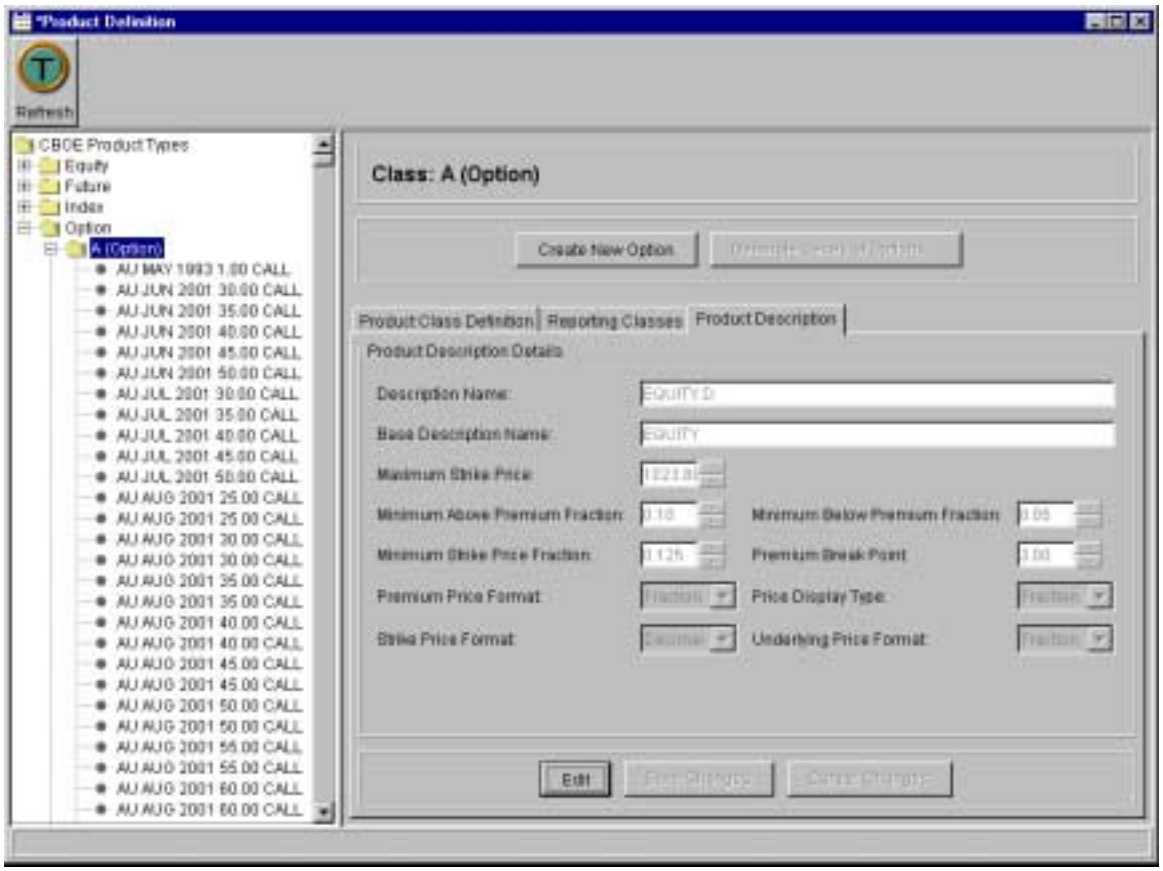
5.4.2.3.2.1 Fields

The table below lists the fields and input field types required for each type of product.

Field	Input Type	Equity	Future	Index	Option
Reporting Class Symbol	text	x	x	x	x
Listing State	drop-down	x	x	x	x
Activation Date	date spinner	x	x	x	x
Inactivation Date	date spinner	x	x	x	x
Contract Size	integer spinner	x	x	x	x

5.4.2.3.3 Edit Product Description

The user interface below depicts the input fields required to edit Product Descriptions.



- Edit Product Description Information
  - This input screen shows the details for the Product Description that was selected in the Product Class Definition screen. Editing this information will impact all other Product Classes that refer to the same Product Description.
  - Information for a Product Description

- Description Name
- Base Description Name
- Maximum Strike Price
- Minimum Above Premium Fraction
- Minimum Below Premium Fraction
- Maximum Strike Price Fraction
- Premium Break Point

5.4.2.3.3.1 Fields

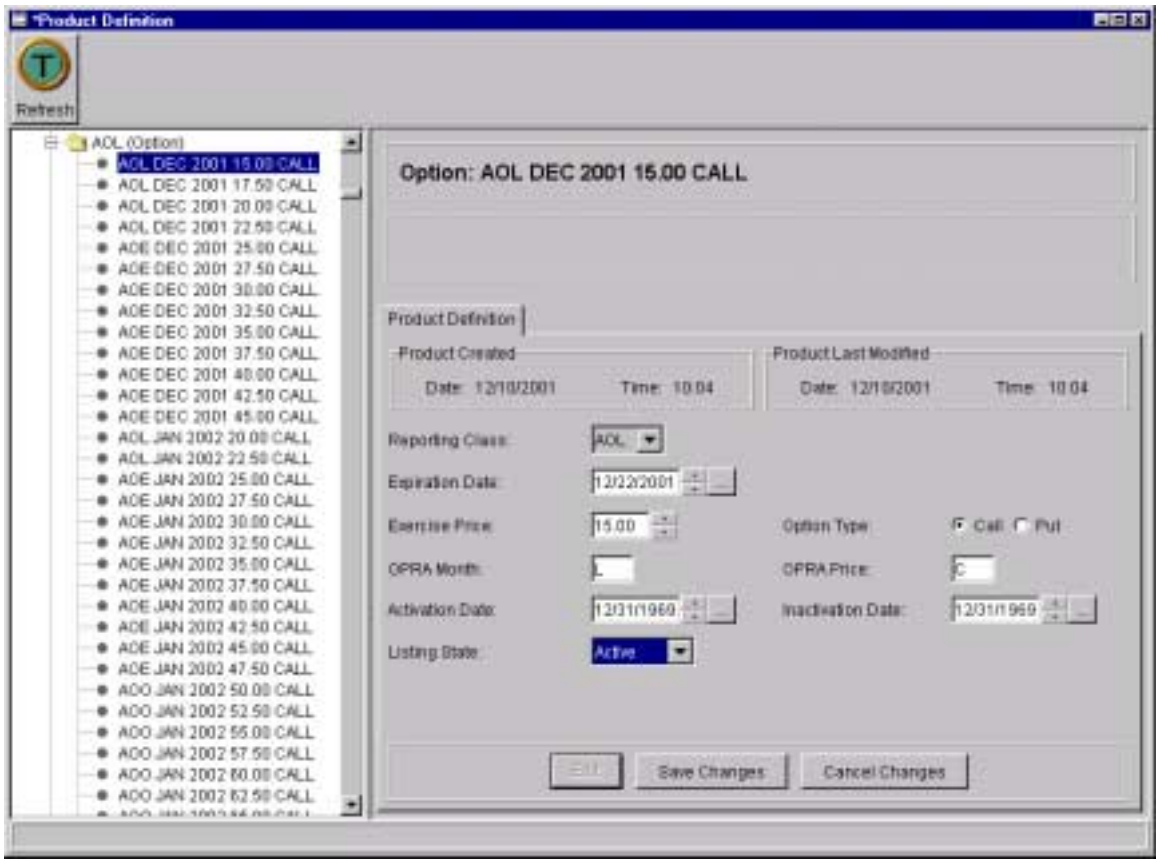
The table below lists the fields and input field types required to edit the Product description.

Field	Input Type	Equity	Future	Index	Option
Description Name	text	x	x	x	x
Base Description Name	text	x	x	x	x
Maximum Strike Price	decimal spinner	x	x	x	x
Minimum Above Premium Fraction	decimal spinner	x	x	x	x
Minimum Below Premium Fraction	decimal spinner	x	x	x	x
Maximum Strike Price Fraction	decimal spinner	x	x	x	x
Premium Break Point	decimal spinner	x	x	x	x
Premium Price Format	drop-down	x	x	x	x
Price Display Type	drop-down	x	x	x	x
Strike Price Format	drop-down	x	x	x	x
Underlying Price Format	drop-down	x	x	x	x

5.4.2.4 Define Product

5.4.2.4.1 Edit Product Definition

The user interface below depicts the information required to define/edit a Product.



- Select the new Product's parent Product Class
- Provide information for the new Product
  - Information for an Equity Product
    - Product Symbol
    - Company Name
    - Activation Date
    - Inactivation Date
    - Listing State
  - Information for a Future Product
    - Expiration Date
    - Activation Date
    - Inactivation Date
    - Listing State
  - Information for an Option Product
    - Reporting Class
    - Expiration Date
    - Exercise Price
    - Option Type
    - OPRA Month Code
    - OPRA Price Code
    - Activation Date
    - Inactivation Date
    - Listing State

5.4.2.4.1.1 Fields

The table below lists the fields and input field types required to edit the new Product.

Field	Input Type	Equity	Future	Index	Option
Reporting Class	drop-down				x
Expiration Date	date spinner		x		x
Exercise Price	Decimal spinner				x
Option Type	radio button				x
OPRA Month	text field				x
OPRA Price	text field				x
Activation Date	date spinner	x	x	x	x
Inactivation Date	date spinner	x	x	x	x
Listing State	drop-down	x	x	x	x
Product Symbol	Text	x		x	
Company Name	Text	x			
Description	Text			x	

5.4.2.5 Delete Product Class

Product Classes can not be directly deleted from the system. Instead, the Product Class must be set to "Inactive" state. All inactive classes will be periodically purged from the system.

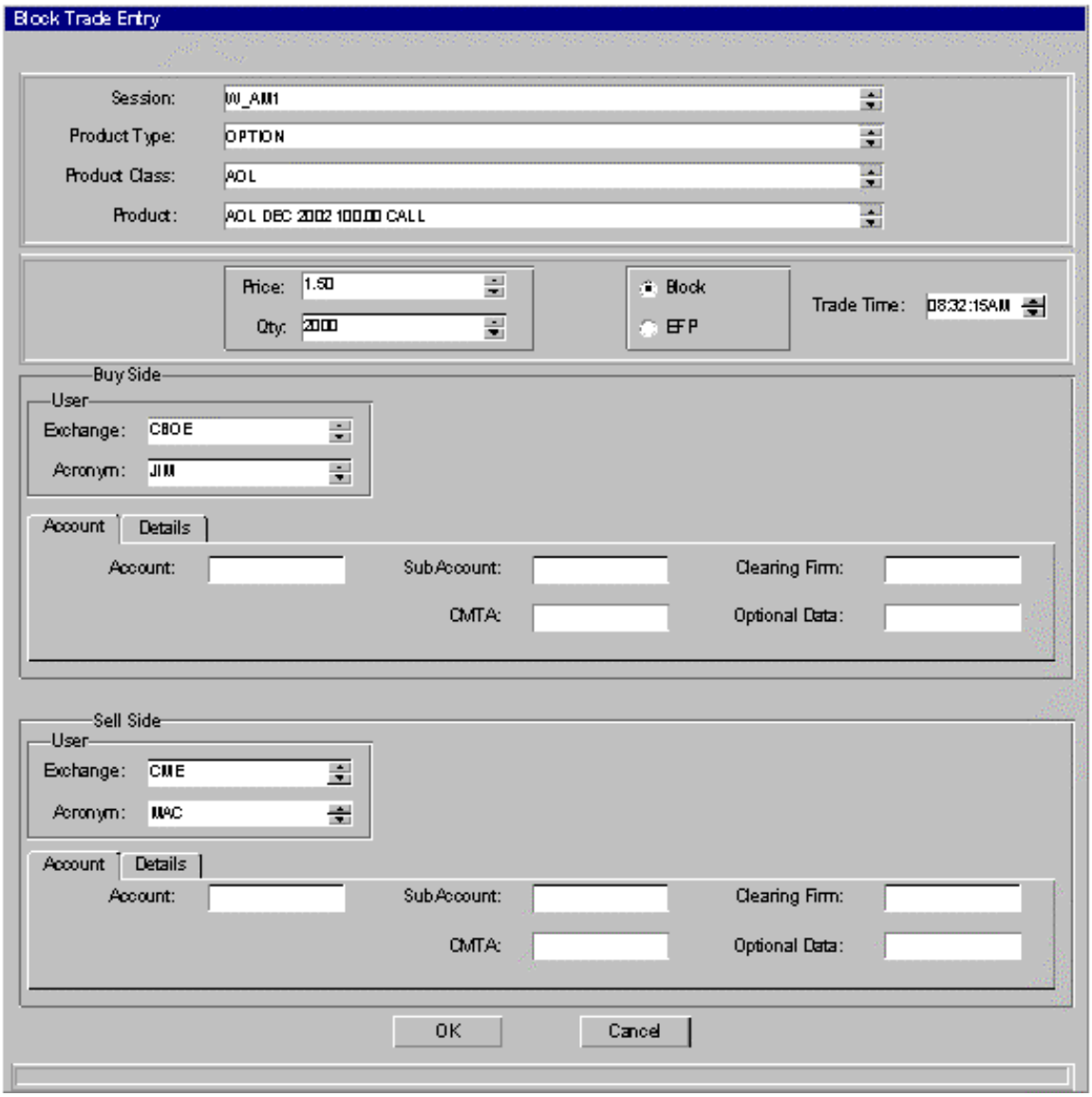
5.4.2.6 Delete Product

Products can not be directly deleted from the system. Instead, the Product must be set to “Inactive” state. All inactive products will be periodically purged from the system.

5.4.3 Block Trading and EFPs

ONE will permit Block Trading and Exchange for Physicals (EFP). The GUI screens depicted below will be used to support this effort.

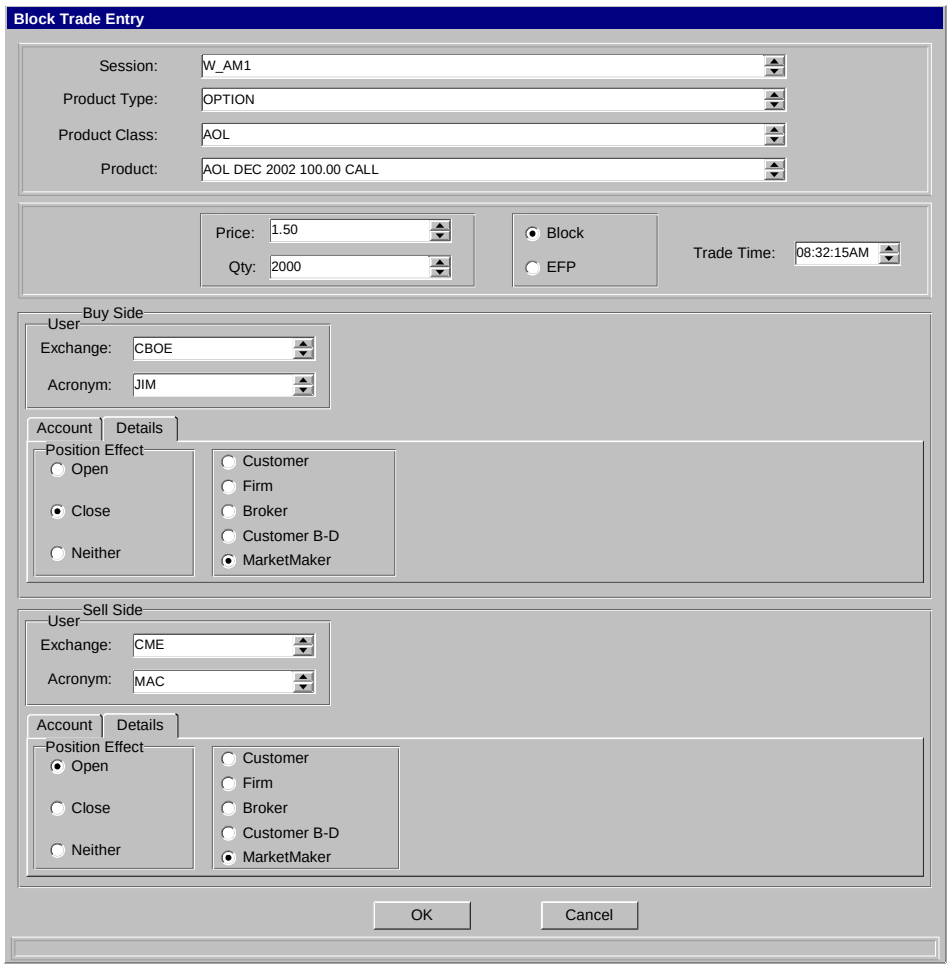
5.4.4 Block Trade Entry Account Tab Displayed



To enter Block Trade information:

- select the session, product type, product class and product from the dropdown lists
- enter the price and quantity
- select Block Trade by selecting the appropriate radio button
- enter the user information for the buy side and sell side
- to submit the information, click ok
- to cancel the request, click cancel

5.4.5 Block Trade Entry with Details Tab Displayed



To enter EFP information:

- select the session, product type, product class and product from the dropdown lists
- enter the price and quantity
- select EFP by selecting the appropriate radio button
- enter the user information for the buy side and sell side
- to submit the information, click ok
- to cancel the request, click cancel

5.4.5.1 Fields

Field Name	Struct	Required	User Entry	Default Value
Quantity	TradeReportStruct	Yes	Yes	
Price	TradeReportStruct	Yes	Yes	
Session name	TradeReportStruct	Yes	Yes	
Product key	TradeReportStruct	Yes	Yes	
Trade source	TradeReportStruct	Yes	No	“Manual”
Trade id	TradeReportStruct	NO	No	0
Trade type	TradeReportStruct	Yes	No	BLOCK from

Field Name	Struct	Required	User Entry	Default Value
				TradeTypes const
Bustable	TradeReportStruct	Yes	No	TRUE
Business day	TradeReportStruct	NO	No	Curr. Date
Time traded	TradeReportStruct	Yes	Yes	Curr time
Atomic trade id	AtomicTradeStruct	No	No	0
Matched sequence number	AtomicTradeStruct	No	No	0
Active	AtomicTradeStruct	No	No	FALSE
Entry time	AtomicTradeStruct	No	No	Curr time
Entry type	AtomicTradeStruct	No	No	“”
Last update time	AtomicTradeStruct	No	No	
Last entry type	AtomicTradeStruct	No	No	
Quantity	AtomicTradeStruct	Yes	No	Equals to row #1
Session name	AtomicTradeStruct	Yes	No	= row #3
Buyer origin type	AtomicTradeStruct	Yes	Yes	MM, Cust, Broker
Buyer firm branch	AtomicTradeStruct	No	No	“”
Buyer firm branch sequence number	AtomicTradeStruct	No	No	“”
Buyer CMTA	AtomicTradeStruct	Yes	Yes	
Buyer CMTA exchange	AtomicTradeStruct	Yes	Yes	
Buyer Correspondent id	AtomicTradeStruct	No	No	
Buyer Position effect	AtomicTradeStruct	Yes	Yes	
Buyer Account acronym	AtomicTradeStruct	Yes	Yes	
Buyer Subaccount	AtomicTradeStruct	Yes	Yes	
Buyer Broker Acronym	AtomicTradeStruct	Yes	Yes	User acronym
Buyer Broker exchange	AtomicTradeStruct	Yes	Yes	
Buyer Originator	AtomicTradeStruct	Yes	Yes	= row #20
Buyer Originator exchange	AtomicTradeStruct	Yes	Yes	
Buyer Firm number	AtomicTradeStruct	Yes	Yes	Clearing firm
Buyer Firm exchange	AtomicTradeStruct	Yes	Yes	
Buyer Optional data	AtomicTradeStruct	No	Yes	“”
Buyer Orderorquoute key	AtomicTradeStruct			
Buyer Order or quote	AtomicTradeStruct			
Reinstatable for buyer	AtomicTradeStruct			
Seller origin type	AtomicTradeStruct			
Seller firm branch	AtomicTradeStruct			
Seller firm branch sequence number	AtomicTradeStruct			
Seller CMTA	AtomicTradeStruct			

Field Name	Struct	Required	User Entry	Default Value
Seller CMTA exchange	AtomicTradeStruct			
Seller Correspondent id	AtomicTradeStruct			
Seller Position effect	AtomicTradeStruct			
Seller Account acronym	AtomicTradeStruct			
Seller Subaccount	AtomicTradeStruct			
Seller Broker Acronym	AtomicTradeStruct			
Seller Broker exchange	AtomicTradeStruct			
Seller Originator	AtomicTradeStruct			
Seller Originator exchange	AtomicTradeStruct			
Seller Firm number	AtomicTradeStruct			
Seller Firm exchange	AtomicTradeStruct			
Seller Optional data	AtomicTradeStruct			
Seller Orderorquoute key	AtomicTradeStruct			
Seller Order or quote	AtomicTradeStruct			
Reinstatable for Seller	AtomicTradeStruct			

5.5 FIX Requirements

5.5.1 Require Sub Account for Order Entry

FIX users will be required to enter the customer account number at the time of order entry. This information will be entered in the sub-account field. The system will validate sub-account information during order entry processing.

5.5.2 Dynamic Book Depth Update

FIX users will be able to retrieve dynamic book depth information by subscribing to the current market by product key and session name. Subscription by class will not be supported. CBOEdirect will support dynamic book depth updates up to n (5) price levels deep.

5.6 Infrastructure Requirements

5.6.1 Capture and Report Performance Statistics

At the end of day, the system will capture and report performance statistics. The time required for CBOE*direct* to process orders must average .5 seconds or less, with a standard deviation of not more than 0.2 seconds. Performance is measured from the point in time that an order or quote enters the front-end processor, through processing, to the point in time when a trade report or other acknowledgement with respect to the order or quote exits the front-end processor.

CBOE must calculate the average speed and the standard deviation on a daily basis. CBOE must also calculate an accumulated average speed and standard deviation for the current week.

The system will provide daily statistics that show by Session, by Transaction Type (Order or Quote), by Minute the mean response time and the transaction rate for that date.

Reports generated by the system should provide a graphical representation with the transaction rate along the x axis and the response time along the y axis. There will be two graphs per session; one for orders, the other for quotes.

6 COPP Requirements

6.1 Specification For Interface Between OPRA Processor and Vendors for Single Stock Futures

6.1.1 Category Codes

New Category Codes	Message
m	Stock Futures Last Sale
n	Stock Futures Open Interest
p	Stock Futures End Of Day Summary
r	Stock Futures Quote With Size
s	Stock Futures Settlement Price

6.1.2 Message Header

PARTICIPANT IDENTIFICATION	1 alpha character
RETRANSMISSION REQUEST	1 alpha character
MESSAGE IDENTIFICATION	2 alpha characters 1 = Message Category 1 = Message Type
MESSAGE SEQUENCE NUMBER	8 numeric characters
TIME	6 numeric characters
	TOTAL = 18 characters

6.2 Sample Record Layout

6.2.1 Stock Futures Last Sale

Data Field	Category m
Futures Symbol	6
Reserved	1
Expiration Date	1
Year	1
Reserved	9
Volume	6

Futures Price Denominator Code	1
Futures Price	8
Reserved	2
	35

6.2.2 Stock Futures Quote with Size

Data Field	Category r
Futures Symbol	6
Reserved	1
Expiration Date	1
Year	1
Reserved	9
Futures Price Denominator Code	1
Bid Quote	8
Bid Size	5
Ask Quote	8
Ask Size	5
Reserved	2
	47

6.2.3 Stock Futures Open Interest

Data Field	Category n
Futures Symbol	6
Reserved	1
Expiration Date	1
Year	1
Reserved	9
Open Interest	7
Reserved	2
	27

6.2.4 Stock Futures End Of Day Summary

Date Field	Category p
Futures Symbol	6
Reserved	1
Expiration Date	1
Year	1
Reserved	9
Volume	6
Open Interest	7
Futures Price Denominator Code	1
Open	8
High	8
Low	8
Last	8
Net Change Indicator	1 (+, - or 0)
Net Change	8
Underlying Price Denominator Code	1

Underlying Stock Price	11
Bid Quote	8
Ask Quote	8
Reserved	2
	103

6.2.5 Stock Futures Settlement Price

Data Field	Category s
Futures Symbol	6
Reserved	1
Expiration Date	1
Year	1
Reserved	15
Settlement Price Denominator Code	1
Settlement Price	8
Reserved	2
	35

7 COMPASS Requirements

7.1 Support Futures Format

7.1.1 FUTURES ORDER/CANCEL/CANCEL REPLACE Format

Although some fields may contain optional data, most lines are required to be sent, either with data, or as blanks. This is indicated by the "(required)" after the line numbers. Individual field requirements are indicated in the format descriptions.

Line 0 (Optional): FIRM MNEMONIC or CORRESPONDENT ACRONYM

1-4 alpha

Member firms who share communications facilities must enter the firm mnemonic on Line 0. This will enable identification of the firm originating the message. Firms who do not share any communications facilities may omit Line 0.

Line 1(Required): BRANCH/SEQUENCE NUMBER

1-3 alphas for BRANCH, followed by a blank space, followed by 1-4 digits for SEQUENCE NUMBER

Field 1 (Required): Identifies the 1-3 alpha branch office code used by the firm.

A blank must follow the BRANCH code.

Field 2 (Required): Identifies the 1-4 numeric sequence number.

The combination of MNEMONIC, BRANCH, SEQUENCE NUMBER and DATE is used by the CBOE as the reference number and must be unique on each order.

1A (Required): Either EXCHANGE IDENTIFIER or BLANK

If used by the entering firm as an EXCHANGE IDENTIFIER, the field will be edited for 1-4 alphas. CBOE will not use this information for routing purposes.

Line 1B (Required): SUMMARY

The format must be exactly as follows: The letter "O", "C", "R" (Required); followed immediately by a blank space, followed immediately by the literal "FUTURES".

Field 1(Required): Message code

Position 1 (Required): Must contain one of the following letters:
"O" if a new **Order**
"C" if a straight **Cancel** of a previously entered order
"R" if a cancel/**Replace** of a previously entered order

(Required): A blank space will follow the message code.

Field 2 (Required): Must be the literal "FUTURES"

Blank space will follow field2 "FUTURES"

Field 3 (Optional): SESSION INDICATOR

Session Indicator, 1 to 16 Alpha/Numeric field if nothing is sent this field will be blank.

Line 2 (Required): TRANSACTION TYPE The literal "BUY" or "SL

On straight cancel requests, the literal "CXL" must be entered in the first field, prior to "BUY" or "SL".

Line 3 (Required): Basic Order Data QUANTITY/SYMBOL/MONTH/PRICE

Field 1 (Required): Identifies the QUANTITY on the order. The maximum value is 9999999. Leading zeros are not required.

A blank space must be entered after the QUANTITY.

Field 2 (Required): Identifies the 1–6 alpha/numeric SECURITY SYMBOL . Will be edited for validity.

Examples: "IBM"
"IBM2"
"IBM2V"

A blank space is required after the SECURITY SYMBOL field.
Field 3 (Required): Identifies the 3-alpha EXPIRATION MONTH
4-digit EXPIRATION YEAR (Required), it must immediately follow EXPIRATION MONTH, with no blank space.

A blank space is required after MONTH/YEAR.

Field 4 (Required): Identifies the order PRICE. "MKT" for a market order. A limit price must be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable). "MOC" for a market-on-close order.

Field 5 (Optional): PRICE QUALIFIER or CONTINGENCY. Valid entries are:
"STP" = Stop or Stop Loss, or Stop Limit *
"AON" = All-or-None
"FOK" = Fill-or-Kill
"IOC" = Immediate or Cancel

A blank space is required after Field 5 if "STP" is entered and the order is a STOP LIMIT.

*Field 6 (Required for Stop Limit): Identifies the PRICE on the STOP LIMIT order. "STP" must be in field 5. If a STOP LIMIT order than the PRICE on the limit must be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable).

A blank space is required after the STOP LIMIT PRICE in field 6.

*Field 7 (Required for Stop Limit): Must contain the literal "LMT".

Line 3A (Required): TIME-IN-FORCE

Field 1 (Required): Identifies the TIME-IN-FORCE. Valid entries are:
"DAY" = Good for the day only
"GTC" = Good-til-Canceled
"DAA" = Good for the morning trading session only (Extended Trading Hours)

Line 3B Identifies the POSITION, ORIGIN and COVERED/UNCOVERED

Field 1 (Optional): Identifies whether the transaction is OPENing or CLOSing. If left blank "CLOSE" is assumed.
"OPEN" = Open transaction
"CLOSE" = Closing transaction

A blank space is required after POSITION

Field 2 (Required): ORIGIN

Field 1 (Required): Identifies the ORIGIN. Valid entries are:
"B" = Broker/dealer
"CUST" = Customer
"D" = DPM
"F" = Firm
"M" = Market maker

A blank space is required after ORIGIN

Field 3 (Optional): Identifies whether the transaction is COVERED or UNCOVERED.
"CVRD" = Covered transaction
"UNCVRD" = Uncovered transaction

Line 3C Used only for cancel/replace orders.

In the "short-form" cancel/replace format, this line must contain the literal "CXL" followed by a blank space, followed immediately by the price on the order being canceled.

In the "long-form" cancel/replace format, this line begins the entire text of the order being canceled, and must be used in conjunction with lines 3D, 3E and 3F.

Field 1(required for cancels): Contains the literal "CXL".

A blank space is required after Field 1.

Field 2: Contains either the PRICE on the order being canceled (if short-form), or "BUY"/"SL" (if long-form).

Line 3D (required on long-form): Identifies the TEXT of the order being canceled
VOLUME/SYMBOL/MONTH/PRICE/QUALIFIER/CONTINGENCY (same format as line 2A).

Line 3E (Required on long-form): Identifies the TIME-IN-FORCE on the order being canceled
(same format as line 3A).

Line 3F (Required on long form): Identifies the ORIGIN/OPEN/COVERED on the order being canceled
(same format as line 3B)

Line 4 (Required): ACCOUNT NUMBER
Up to 16 alpha/numeric characters. This information is not edited and is simply passed along on the order.

Line 4A (Required): REFERENCE or BLANK
This information is only applicable for cancels and cancel/replaces. If the order is not a cancel or cancel/replace, then this line must still be sent as a BLANK. If the message is a cancel or cancel/replace, then the REFERENCE information must be entered exactly as follows:

Field 1: Must contain the literal "RE".
A blank space must follow "RE".

Field 2: Must contain the 1-3 alpha BRANCH of the order being canceled.
A blank space must follow the BRANCH.

Field 3: Must contain the 1-4 digit SEQUENCE NUMBER of the order being canceled. A forward slash ("/") must be entered immediately after the SEQUENCE NUMBER.

Field 4: Must contain the entry date of the order being canceled. The format is "mmddyyyy", where mm is the 2-digit month; dd is the 2-digit day; and yyyy is the year (yyyy is optional).

Line 4B (Required): CMTA NUMBER or BLANK.
If no CMTA information is included on the order, then this line must still be sent as a BLANK. If CMTA is included, it must be sent in the following format:

Field 1(Required): Must contain the literal "CMTA".

A blank space must be entered after the letters "CMTA".

Field 2 (Required): One letter Alpha for Exchange, "O" or "M".

Field 3 (Required): 3 digit OCC or CME account into which the trade is to clear (leading zeros required).

Line 4C (Required): MISCELLANEOUS DATA or BLANK
If no MISCELLANEOUS DATA is included on the message, then this line must still be sent as a BLANK. If MISCELLANEOUS DATA is included, the first 32 characters will be passed along .

Line 5 (Required): Firm-specific TRAILER information

Example 1 – Futures Order:

Line 0:	JCB' (correspondent firm mnemonic)
Line 1:	XJ 1354' (branch/sequence)
Line 1A:	' (optional exchange identifier or blank)
Line 1B:	O FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES". Field 3 is session indicator or blank space)
Line 2:	BUY'
Line 3:	10 IBM OCT2001 105.60'
Line 3A:	DAY'
Line 3B:	OPEN CUST CVRD'
Line 4:	123-45678-90A' (account number or blank)
Line 4A:	' (reserved for RE information on cxls and cx/res)
Line 4B:	CMTA O382' (CMTA number or blank)
Line 4C:	CFM 10 IBM OCT 105.60'(miscellaneous optional data or blank)
Line 5:	TRAILER

Example 2A – Futures Cancel/Replace (short-form):

Line 0:	JCB' (correspondent firm mnemonic)
Line 1:	XJ 1365' (branch/sequence)
Line 1A:	' (optional exchange identifier or blank)
Line 1B:	R FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES". Field 3 is session indicator or blank space)
Line 2:	BUY'
Line 3:	10 IBM OCT2001 105.80'
Line 3A:	DAY'
Line 3B:	OPEN CUST CVRD'

Line 3C: CXL 105.60’ (price of the order being canceled)
Line 4: 123-45678-90A’ (account number or blank)
Line 4A: RE XJ 1354/05032001’ (reference branch/sequence/entry date of order being canceled)
Line 4B: CMTA O382’ (CMTA number or blank)
Line 4C: CFM CXL 105.60’ (miscellaneous optional data or blank)
Line 5: TRAILER

Example 2B – Futures Cancel/Replace (Long-form):

Line 0: JCB’ (correspondent firm mnemonic)
Line 1: XJ 2189’ (branch/sequence)
Line 1A: ‘ (optional exchange identifier or blank)
Line 1B: R FUTURES JVAM1’ (summary, a blank space will go between message code and the word “FUTURES”. Field 3 is session indicator or blank space)
Line 2: BUY’
Line 3: 10 IBM OCT2001 106.00’
Line 3A: DAY’
Line 3B: OPEN CUST CVRD’
Line 3C: CXL BUY’
Line 3D: 10 IBM OCT2001 105.80’
Line 3E: DAY’
Line 3F: OPEN CUST CVRD’
Line 4: 123-45678-90A’ (account number or blank)
Line 4A: RE XJ 1365/05032001’ (reference branch/sequence/entry date of order being canceled)
Line 4B: CMTA O382’ (CMTA number or blank)
Line 4C: CFM CXL IBM 105.80’ (miscellaneous optional data or blank)
Line 5: TRAILER

Example 3 - Futures Straight Cancel:

Line 0: JCB’ (correspondent firm mnemonic)
Line 1: XJ 2197’ (branch/sequence)
Line 1A: ‘ (optional exchange identifier or blank)
Line 1B: C FUTURES’ (summary, a blank space will go between message code and the word “FUTURES”)
Line 2: CXL BUY’
Line 3: 10 IBM OCT2001 106.00’
Line 3A: DAY’
Line 3B: OPEN CUST CVRD’
Line 4: 123-45678-90A’ (account number or blank)
Line 4A: RE XJ 2189/05032001’ (reference branch/sequence/entry date of order being canceled))
Line 4B: CMTA O382’ (CMTA number or blank)
Line 4C: CFM CXL 106.00’ (miscellaneous optional data or blank)
Line 5: TRAILER

7.1.2 Futures Execution Report Format

Line -1: (Required) HEADER Firm-specific header information

Line : (Optional) FIRM MNEMONIC or CORRESPONDENT ACRONYM
1-4 alpha
If entered on the original order line 1A, this information will be returned on the report. If not entered on the original order, line 0 will be omitted.

Line 1: (required) BRANCH/SEQUENCE NUMBER
Line 1 of the report contains the branch office code and sequence number of the executed order.

Field 1(Required) Contains the 1-3 alpha branch office code used by the firm.

A blank will follow the BRANCH code.

Field 2:(Required) Contains the 1-4 numeric sequence number.

Line 1A: (Required) SUMMARY
The format will be exactly as follows: The letter "F", "U", "N" (Required); followed immediately by a blank space, followed immediately by the literal "FUTURES".

Field 1: (Required) Message code
Position 1: Will contain one of the following letters:

"F" if a **Fill Report**

(Required): A blank space will follow the message code.

Field 2: (Required) Will be the literal "FUTURES".

A blank space will follow

Field 3: (Optional) Session Indicator ,if sent on original order
Session Indicator or blank space if one isn't provided

Line 2: (Required) TRANSACTION TYPE (Past tense on fills)

Contains one of the following:
For Fill Reports:
"BOT" – Bought
"SLD" – Sold

For Cancel Reports:
"BUY"
" SL"

Line 3: (Required) Execution Data QUANTITY/SYMBOL/MONTH/PRICE

Field 1: (Required) Contains the QUANTITY executed. The maximum value is 9999999. Leading zeros are not required.

(Required) A blank space will be entered after the QUANTITY.

Field 2:(Required) Will contain the 1 - 6 alpha/numeric SECURITY SYMBOL on the original order.

Examples: "IBM"
"IBM2"
"IBM2V"

(Required) A blank space will be sent after the SECURITY SYMBOL field.

Field 3: (Required) Identifies the 3-alpha EXPIRATION MONTH from the original order.
4-digit EXPIRATION YEAR (Required) sent on the original, it will immediately follow EXPIRATION MONTH, with no blank space.

(Required) A blank space will be sent after MONTH/YEAR.

Field 4: (Required)Identifies the EXECUTION PRICE. Will be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable).

Line 4: Basic Order Information

Field 1: (Required) The word 'ON' will always be in this field.

(Required) A blank space will be sent after Field 1.

Field 2: ORIGINAL ORDER PRICE. Will be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable) or "MKT" if original order price was "MKT".

(Required) A blank space will be sent after ORIGINAL ORDER PRICE.

Field 3: Additional PRICE QUALIFIERS from Line 3 on original order.
The literal "LMT" if the original order price was a limit.
The literal "STP" if the original order was a Stop or Stop Limit.

Field 4: Contains the PRICE on the STOP LIMIT order. "STP" must be in field 3. If a STOP LIMIT order than the original PRICE on the limit will be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable).

A blank space will be sent after the STOP LIMIT PRICE in field 4.

Field 5: Will contain the literal "LMT" if the original order was a STOP LIMIT.

Line 4A: (Required) FILLS, LVS, Contingency and TIME-IN-FORCE

Field 1: Leaves volume will be included if the entire order is not executed. For execution reports, this field contains either the term "LVS" followed by a space and a numeric field, indicating the quantity remaining to be executed, or the term "FILLS" indicating the completing execution on an order. If the order is completely filled this field will be omitted.

A blank space is required after Field 1.

Field 2: Contains "AON" if the original order was an all-or-none. If included on the original order, then the information will be returned on the report.
A blank space is required after Field 2.

Field 3: Contains the TIME IN FORCE included on the original order.

(Required) A blank space is required after TIME IN FORCE.

Line 4B: Identifies the POSITION, ORIGIN and COVERED/UNCOVERED
Field 1: (Required)Contains ORIGIN from the original order.
A blank space will be sent after origin.

Field 2:(Required) Contains POSITION from the original order.
A blank space will be sent after origin.

Field 3: (Optional) COVERED/ UNCOVERED
This field will be sent if on the original order. ("CVRD" or "UNCVRD")

Line 4C:(Required) ACCOUNT INFORMATION or BLANK
Contains account information from original order, or will be sent as a blank

Line 5: (Required) CMTA NUMBER or BLANK
If no CMTA information is included on the original order, then this line must still be sent as a blank. If CMTA is included, it will be sent in the following format:

Field 1: Will contain the literal "CMTA".

A blank space will be entered after the letters "CMTA".

Field 2: 3 digit OCC account into which the trade is to clear (leading zeros required).

Line 5A: (Required) MISCELLANEOUS DATA or BLANK
If included on Line 4C on the original order then the MISCELLANEOUS DATA is included on the report. If not, then this line will still be sent as a blank.

Line 6: (Required) CLEARING DATA
Field 1: Executing Broker of 1 – 3 alpha characters will be sent.

A blank space will be sent after field 1.

Field 2: (Required) CONTRA INFORMATION
Contra-Firm Mnemonic of 1 – 4 alpha characters will be sent; followed by the 1-7 numeric quantity executed; followed by a slash ("/"); followed by the 1-3 alpha acronym of the contra broker on the trade, followed by a space and the time of execution (military time).

A space will separate each set of contra information.

Example 1– Futures Order:

Line 0:	JCB' (correspondent firm mnemonic)
Line 1:	XJ 1333' (branch/sequence)
Line 1A:	' (optional exchange identifier or blank)
Line 1B:	O FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES" Field 3 is session indicator or blank space))
Line 2:	BUY'
Line 3:	10 IBM OCT2001 105.60'
Line 3A:	DAY'
Line 3B:	OPEN CUST CVRD'
Line 4:	123-45678-90A' (account number or blank)
Line 4A:	' (reserved for RE information on cxls and cx/res)

Line 4B: CMTA O382’ (CMTA number or blank)
Line 4C: CFM 10 IBM OCT 105.60’(miscellaneous optional data or blank)
Line 5: TRAILER

Example 4 – Futures Execution Report Formats (Fill Report):
Line -1: HEADER
Line 0: JCB’ (correspondent firm mnemonic)
Line 1: XJ 1333’ (branch/sequence)
Line 1A: F FUTURES JVAM1’ (summary, a blank space will go between message code and the word “FUTURES” Field 3 is session indicator or blank space))
Line 2: BOT’
Line 3: 5 IBM OCT2001 100.60’
Line 4: ON 105.60’
Line 4A: LVS 5 DAY
Line 4B: OPEN CUST CVRD
Line 4C: 123-45678-90A’ (account number or blank)
Line 5: CMTA O382’ (CMTA number or blank)
Line 5A: CFM 10 IBM OCT 105.60’(miscellaneous optional data or blank)
Line 6: XXL PW10/ABC 1321

7.1.3 Futures UR OUT or Cancel Report Format

Line –1(Required) HEADER Firm-specific header information

Line 0: (Optional) FIRM MNEMONIC or CORRESPONDENT ACRONYM or BLANK
1-4 alpha
If entered on the original order line 1A, this information will be returned on the report. If not entered on the original order, line 0 will be omitted.

Line 1: (Required)BRANCH/SEQUENCE NUMBER
Line 1 of the report contains the branch office code and sequence number of the executed order.

Field 1: Contains the 1-3 alpha branch office code used by the firm.

A blank will follow the BRANCH code.

Field 2: Contains the 1-4 numeric sequence number.

Line 1A: (Required) SUMMARY
The format will be exactly as follows: The letter “F”, “U”, “N” (Required); followed immediately by a blank space, followed immediately by the literal “FUTURES”.

Field 1: Message code
Position 1: Will contain one of the following letters:

“U” if a **UR OUT report**

(Required): A blank space will follow the message code.

Field 2: Will be the literal “FUTURES”.

A blank space will follow

Field 3: (Optional) Session Indicator

Session Indicator or blank space if one isn't provided

Line 2: (Required) TRANSACTION TYPE (Past tense on cancel orders U R out reports)

For Cancel Reports:
"BUY" – buy
"SL" – sell

Line 3: (Required) Basic Order Information QUANTITY/SYMBOL/MONTH/PRICE/CONTINGENCY

Field 1: Contains the QUANTITY on the original order. The maximum value is 9999999. Leading zeros are not required.

A blank space will be entered after the QUANTITY.

Field 2: Will contain the 1 - 6 alpha/numeric SECURITY SYMBOL on the original order.

Examples: "IBM"
"IBM2"
"IBM2V"

A blank space will be sent after the SECURITY SYMBOL field.

Field 3: Identifies the 3-alpha EXPIRATION MONTH from the original order.
4-digit EXPIRATION YEAR (Required) sent on the original, it will immediately follow EXPIRATION MONTH, with no blank space.

A blank space will be sent after MONTH/YEAR.

Field 4: Identifies the ORIGINAL PRICE. Will be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable).

Field 5: PRICE QUALIFIER or CONTINGENCY
If included on original order.

A blank space will be sent after Field 5, if included.

Field 6: PRICE of STOP LIMIT
If included on original order.

Field 7: "LMT" if included on original order.

Line 3A: (Required) UR OUT and Quantity

Field 1: For cancel reports, this field will contain the literal "UR OUT" it may also contain the term "TLC.

A blank space will follow Field 1.

Field 2: This field will contain the quantity that was canceled.

Line 3B: (Required) LVS, TLC, FILLS, TIME IN FORCE or BLANK

Field 1: If volume remains "LVS" will appear on the cancel reports. If the order being cancelled had the original

volume changed in anyway, the term “FILLS” will appear indicating the completing transaction.

A blank space will follow “LVS” in field 1.

Field 2: Contains Leaves volume if the entire order is not canceled followed by a blank space and the TIME IN FORCE.

Line 4: Identifies the POSITION, ORIGIN and COVERED/UNCOVERED
Field 1: (Required) Contains POSITION of the original order.
A blank space will be sent after origin.

Field 2: (Required) Contains ORIGIN from the original order.

A blank space will be sent after origin.

Field 3: (Optional) COVERED/ UNCOVERED
This field will be sent if on the original order. (“CVRD” or “UNCVRD”)

Line 4A: (Required) Account Information
Contains account information from original order, or will be sent as a blank.

Line 5: (Required) CMTA NUMBER
If no CMTA information is included on the original order, then this line must still be sent as a blank. If CMTA is included, it will be sent in the following format:

Field 1: Will contain the literal “CMTA”.

A blank space will be entered after the letters “CMTA”.

Field 2: 3 digit OCC account into which the trade is to clear (leading zeros required).

Line 5A: (Required) MISCELLANEOUS DATA or BLANK
If included on Line 4C on the original order then the MISCELLANEOUS DATA is included on the report. If not, then this line will still be sent as a blank.

Example 1– Futures Order:

Line 0:	JCB’ (correspondent firm mnemonic)
Line 1:	XJ 1335’ (branch/sequence)
Line 1A:	‘ (optional exchange identifier or blank)
Line 1B:	O FUTURES JVAM1’ (summary, a blank space will go between message code and the word “FUTURES” Field 3 is session indicator or blank space))
Line 2:	BUY’
Line 3:	10 IBM OCT2001 105.60 IOC’
Line 3A:	DAY’
Line 3B:	OPEN CUST CVRD’
Line 4:	123-45678-90A’ (account number or blank)
Line 4A:	‘ (reserved for RE information on cxls and cx/res)
Line 4B:	CMTA O382’ (CMTA number or blank)
Line 4C:	CFM 10 IBM OCT 105.60’(miscellaneous optional data or blank)
Line 5:	TRAILER

Example 5 – Futures Cancel Report Formats (UR OUT):

Line -1:	HEADER
Line 0:	JCB’ (correspondent firm mnemonic)
Line 1:	XJ 1335’ (branch/sequence)

Line 1A: U FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES" Field 3 is session indicator or blank space))
Line 2: BUY'
Line 3: 10 IBM OCT2001 105.60 IOC'
Line 3A: UR OUT 10
Line 3B: (blank)
Line 4: OPEN CUST CVRD
Line 4A: 123-45678-90A' (account number or blank)
Line 5: CMTA O382' (CMTA number or blank)
Line 5A: CFM 10 IBM OCT 105.60'(miscellaneous optional data or blank)

7.1.4 Futures Nothing Done Report Format

Line -1: (Required) HEADER Firm-specific header information

Line 0: (Optional) FIRM MNEMONIC or CORRESPONDENT ACRONYM or BLANK
1-4 alpha
If entered on the original order line 1A, this information will be returned on the report. If not entered on the original order, line 0 will be omitted.

Line 1: (Required) BRANCH/SEQUENCE NUMBER
Line 1 of the report contains the branch office code and sequence number of the executed order.

Field 1: Contains the 1-3 alpha branch office code used by the firm.

A blank will follow the BRANCH code.

Field 2: Contains the 1-4 numeric sequence number.

Line 1A: (Required) SUMMARY
The format will be exactly as follows: The letter "F", "U", "N" (Required); followed immediately by a blank space, followed immediately by the literal "FUTURES".

Field 1: Message code
Position 1: Will contain one of the following letters:

"N" if a **NOTHING DONE report**

(Required): A blank space will follow the message code.

Field 2: Will be the literal "FUTURES".

A blank space will follow

Field 3: Session Indicator or blank space if one isn't provided

Line 2: (Required) TRANSACTION TYPE (Past tense on cancel orders U R out reports)

For Nothing Done Reports:
" BUY"
" SL"

Line 3: (Required) Basic Order Information QUANTITY/SYMBOL/MONTH/PRICE/CONTINGENCY

Field 1: Contains the QUANTITY of the NOTHING DONE REPORT. The maximum value is 9999999. Leading
zeros are not required.

A blank space will be entered after the QUANTITY.

Field 2: Will contain the 1 - 6 alpha/numeric SECURITY SYMBOL on the original order.

Examples: "IBM"
"IBM2"
"IBM2V"

A blank space will be sent after the SECURITY SYMBOL field.

Field 3: Identifies the 3-alpha EXPIRATION MONTH from the original order.
4-digit EXPIRATION YEAR (Required) sent on the original, it will immediately follow EXPIRATION MONTH, with no blank space.

A blank space will be sent after MONTH/YEAR.

Field 4: Identifies the ORIGINAL PRICE. Will be entered as up to 4 digits for integer (no leading zeros required), followed by a decimal point and two digits (if applicable).

Field 5: PRICE QUALIFIER or CONTINGENCY
If included on original order.

A blank space will be sent after Field 5, if included.

Field 6: PRICE of STOP LIMIT
If included on original order.

Field 7: "LMT" if included on original order.

Line 3A: (Required) NOTHING DONE and Quantity
For nothing done reports, this field will contain the literal "NOTHING DONE"

Line 3B: (Required) PART OF ORIGINAL QUANTITY or BLANK
Contains the literal "PART OF" If report is for a volume different than original volume followed by a
space.

Line 4: Identifies the POSITION, ORIGIN and COVERED/UNCOVERED

Field 1: (Required) Contains POSITION of the original order.
A blank space will be sent after origin.

Field 2: (Required) Contains ORIGIN from the original order.
A blank space will be sent after origin.

Field 3: (Optional) COVERED/ UNCOVERED
This field will be sent if on the original order. ("CVRD" or "UNCVRD")

Line 4A: (Required) Account Information
Contains account information from original order, or will be sent as a blank.

Line 5: (Required) CMTA NUMBER

If no CMTA information is included on the original order, then this line must still be sent as a blank. If CMTA is included, it will be sent in the following format:

Field 1: Will contain the literal "CMTA".

A blank space will be entered after the letters "CMTA".

Field 2: 3 digit OCC account into which the trade is to clear (leading zeros required).

Line 5A: (Required) MISCELLANEOUS DATA

If included on Line 4C on the original order then the MISCELLANEOUS DATA is included on the report. If not, then this line will still be sent as a blank.

Example 1 – Futures Order:

Line 0: JCB' (correspondent firm mnemonic)
Line 1: XJ 1336' (branch/sequence)
Line 1A: ' (optional exchange identifier or blank)
Line 1B: O FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES" Field 3 is session indicator or blank space))
Line 2: BUY'
Line 3: 20 IBM OCT2001 105.60'
Line 3A: DAY'
Line 3B: OPEN CUST CVRD'
Line 4: 123-45678-90A' (account number or blank)
Line 4A: ' (reserved for RE information on cxls and cx/res)
Line 4B: CMTA O382' (CMTA number or blank)
Line 4C: CFM 10 IBM OCT 105.60'(miscellaneous optional data or blank)
Line 5: TRAILER

Example 6 – Futures Nothing Done Report Format:

Line -1: HEADER
Line 0: JCB' (correspondent firm mnemonic)
Line 1: XJ 1336' (branch/sequence)
Line 1A: N FUTURES JVAM1' (summary, a blank space will go between message code and the word "FUTURES" Field 3 is session indicator or blank space))
Line 2: BUY'
Line 3: 10 IBM OCT2001 105.60'
Line 3A: NOTHING DONE
Line 3B: PART OF 20
Line 4: OPEN CUST CVRD
Line 4A: 123-45678-90A' (account number or blank)
Line 5: CMTA O382' (CMTA number or blank)
Line 5A: CFM 10 IBM OCT 105.60'(miscellaneous optional data or blank)

8 TIPS Requirements

TIPS will require new OPRA-spec messages for Futures. These messages will be an extension of the existing OPRA spec. TIPS will decode the OPRA-spec messages and treat them like stocks in order to support Futures.

9 Market Maker Handheld Requirements

10 Trade Match Requirements

10.1 Create Separate Futures Database

10.1.1 Futures Trades Database Record

		FIELD				
FIELD NAME	***SBT X-REF***	Status	Chg	***BYTES***	***FMT***	***DESCRIPTION***
Executing Firm Exchange	Buyer/Seller Firm Exchange			5	X	CBOE, CME, CBOT
Executing Firm Code	Buyer/Seller Firm	R		5	N	
Transaction ID Match Number		R		7	N	Matched Sequence Number
Transaction ID Buy / Sell		R		1	X	B = buy; S = sell
Transaction ID Allocation Nbr				3	N	Associated with Transaction Seq. Nbr.
Business Date	Business Trade Date			8	X	CCYYMMDD
Execution Trade Date	Execution Trade Date	R		8	X	CCYYMMDD
Execution Trade Time	Execution Trade Time	R		8	X	HHMMSSff
Trade Quantity (Contract Qty)	Trade Quantity	R	L	7	N	
Market Multiplier	Market Multiplier (Contract Size)			7	N	
Trade Price Dollar	Trade Price Dollar	R		6	N	
Trade Price Fraction	Trade Price Fraction	R		4	X	'.50' = 5000
Product Line Type Code	Product Line Code			2	X	from SBT
Security Type Code	Security Type Code			2	X	
Trading Symbol	Trading Symbol	R		12	X	Initially 6 characters max.
Expiration Date	Expiration Date	R		6	X	CCYYMM
Filler				2	X	
Executing Broker Exchange	Buyer/Seller Acronym Exchange			5	X	
Executing Broker Code	Buyer/Seller Acronym	R		5	X	
Contra Firm Exchange	Seller/Buyer Firm Exchange			5	X	CBOE, CME, CBOT
Contra Firm Code	Seller/Buyer Firm	R		5	N	
Contra Broker Exchange	Seller/Buyer Acronym Exchange			5	X	
Contra Broker Code	Seller/Buyer Acronym	R		5	X	
Special Trade Indicator	Special Trade Indicator	O		1	X	B = block trade E = exchange for physical (EFP)

		FIELD				
FIELD NAME	***SBT X-REF***	Status	Chg	***BYTES***	***FMT***	***DESCRIPTION***
Trade Originator Exchange	Buyer/Seller Originator Exchange			5	X	CBOE, CME, CBOT
Trade Originator Code	Buyer/Seller Originator			5	X	
Trade Report UID	SBT Trade Report ID			20	N	big number from SBT
Session Indicator	SBT Session ID			10	X	
filler				5	X	
Account Type (Origin Code)	Buyer/Seller Account Type	R	C, L	1	X	CBOE Codes: C = Customer F = Firm M = Market Maker CME Codes: V = CTI 1/Origin 1 H = CTI 3/Origin 2 E = CTI 1/Origin 2 R = CTI 3/Origin 5 Q = CTI 1/Origin 5 C = CTI 4/Origin 1 F = CTI 2/Origin 2 O = CTI 4/Origin 2 G = CTI 3/Origin 1 T = CTI 4/Origin 5
Open/Close Code	Buyer/Seller Open-Close Indicator	O	C, L	1	X	C = Close Transaction O = Open Transaction space = undetermined
CMTA Firm Status				1	X	space = OK E = error I = invalid relationship P = pending R = released
CMTA Firm Exchange	Buyer/Seller CMTA Firm Exchange		C, L	5	X	CBOE, CME, CBOT

		FIELD				
FIELD NAME	***SBT X-REF***	Status	Chg	***BYTES***	***FMT***	***DESCRIPTION***
CMTA Firm Code	Buyer/Seller CMTA Firm	O	C, L	5	N	
Joint Account	Buyer/Seller Account		C, L	10	X	Q account / joint account
Sub-Account	Buyer/Seller Subaccount	R	C, L	10	X	MM account / customer account
Firm Branch	Buyer/Seller Firm Branch			5	X	
Firm Branch Sequence Number	Buyer/Seller Firm Branch Sequence Number			10	X	
Correspondent ID	Buyer/Seller Correspondent ID			5	X	
Optional Data	Buyer/Seller Optional Data	O	C, L	32	X	
Clearing Account Type				1	X	C, F or M
Clearing Firm Exchange			X	5	X	CBOE, CME, CBOT
Clearing Firm Code			X	5	N	
PTP Firm Exchange				5	X	CBOE, CME, CBOT
PTP Firm Code				5	N	Authority to change trade record
Entry Source				1	X	S = SBT input B = batch input
Entry Date				8	X	CCYYMMDD
Entry Time				8	X	HHMMSSff
Trade Status				1	X	C = cleared trade E = error trade D = deleted trade M = matched trade
Update Status				1	X	U = updated record L = allocated add record
Outbound Network Status				1	X	Space = not sent S = sent U = updated
Filler				5	X	
		TOTAL	=	300		

10.9 Pass Futures Trades to OCC

For the OCC Futures Matched Trade record, numeric data is right justified, zero filled.

Alpha/Numeric data is left justified, blank filled.

Filler fields are maintained by OCC and are reserved for future enhancements.

10.9.1 Futures Matched Trade Record

Position	Field Name	Picture	Comments
01 – 03[1]	Transaction ID	PIC 9(03)	430=Matched Trades
04 – 08	Buy Clearing Member Number	PIC 9(05)	
09 – 09	Buy Clearing Member Account Type	PIC X(01)	C=Customer F=Firm Account M=Market Maker
10 – 13	Buy Floor Trader ID	PIC X(04)	
14 – 18	Buy CMTA Firm	PIC 9(05)	Buy side CMTA firm's OCC Clearing Member Number
19 – 19	Buy Open/Close Indicator	PIC X(01)	O=Open C=Closed
20 – 23	Buy Executing Broker	PIC X(04)	Executing Broker's Acronym
24 – 39	Buy Optional Data	PIC X(16)	Buy Firm's Optional Data
40 – 49	Buy Account Number	PIC X(10)	
50 – 53	Filler Space	PIC X(04)	
54 – 58	Sell Clearing Member Number	PIC 9(05)	
59 – 59	Sell Clearing Member Account Type	PIC X(01)	C=Customer Account F=Firm Account M=Market Maker Account
60 – 63	Sell Floor Trader ID	PIC X(04)	
64 – 68	Sell CMTA Firm	PIC 9(05)	Sell side CMTA firm's Clearing Member Number
69 – 69	Sell Open/Close Indicator	PIC X(01)	O = Open C = Close
70 – 73	Sell Executing Broker ID	PIC X (04)	Executing Broker's Acronym
74 – 89	Sell Optional Data	PIC X (16)	Sell Firm's Optional Data
90 – 99	Sell Account Number	PIC X (10)	
100 – 105	Filler Space	PIC X (03)	
106 – 111[1]	Commodity Code	PIC X (06)	Future or Security Future Trading Symbol
112-114	Filler Space	PIC X(02)	
115 – 116	Expiration Month	PIC 9(02)	Two Digit Month
117 – 118	Filler Space	PIC X (02)	
119 – 122	Expiration Year	PIC 9 (04)	CCYY
123 – 130	Filler Space	PIC X (08)	
131 – 134	Trade Price Dollar Amount	PIC 9(04)	Whole Dollar Portion of Trade Price
135 – 138	Trade Price Decimal Amount	PIC X (04)	Decimal or Decimal Equivalent of Fraction—***Left Justified, trades with no decimal price should have blanksin this field. A price of $3.70 would be entered as 70~~bb~~ where ~~bb~~ represents spaces.***

Position	Field Name	Picture	Comments
139 – 144	Filler Space	PIC X(06)	
145 –151	Trade Quantity	PIC 9(07)	
152 – 159	As Of Date	PIC X(08)	MMDDCCYY, ***may be left blank***
160 – 163	Buy Side Execution Time	PIC X(04)	HHMM or Blank
164 – 167	Sell Side Execution Time	PIC X(04)	HHMM or Blank
168 – 169	Filler Space	PIC X(02)	
170 – 171	Exchange Code	PIC 9(02)	Identity of submitting Exchange 07 = ONE
172 – 173	Exchange Location Code	PIC X(02)	Identity of where the trade was executed. 01 = CBOE 02 = AMEX 03 = PHLX 04 = PSE 06 = NQLX 07 = ONE 08 = ISE ***This is an optional field and may be left blank. Any data entered in this field will not be edited by OCC.***
174 – 175	Exchange Billing Code	PIX X(02)	***This is an optional field and may be left blank. Any data entered in this field will not be edited by OCC.***
176 – 176	Exchange for Physical Indicator	PIC X(01)	Y or N; ***may be left blank***
177 – 183[1, 2]	Exchange Sequence Number	PIC X(07)	
184 – 184	Trade Type	PIC X(01)	" = New Trade B = Busted Trade
185 – 185	Block Trade Indicator	PIC X(01)	Y or N; ***may be left blank***
186 –186	Filler Space	PIC X(01)	
187 – 194	Activity Date	PIC X(08)	MMDDCCYY
195 –250	Filler Space	PIC X(56)	

[1] Required field on a "Bust" transaction.
[2] Exchange generated number for each trade—will be the same for buy and sell sides. This is a mandatory field required for OCC processing as of January 2002. This sequence number must be unique to one and only one trade.

10.9.2 Futures Matched Trade Trailer Record

Position	Field Name	Picture	Comments
01 – 01	Trailer Record ID	PIC X(01)	T = Trailer Literal
02 – 06	Filler Space	PIC X(05)	
07 – 18	Total Number of Records	PIC 9(12)	
19 – 35	Filler Space	PIC X(17)	
36 – 43	Activity Date	PIC X(08)	MMDDCCYY
44 – 45	Filler Space	PIC X(02)	
46 – 47	Exchange Code	PIC 9(02)	07 = ONE
48 – 49	Filler Space	PIC X(02)	
50 – 61	New Trade Record Count	PIC 9(12)	

Position	Field Name	Picture	Comments
62 – 73	New Trade Contract Count	PIC 9(12)	
74 – 85	Busted Trade Record Count	PIC 9(12)	
86 – 87[1]	Batch Submission Number	PIC X(02)	***This field is currently not used and may be left blank.***
88 – 93[1]	Batch Submission Time		HHMMSS ***This field is currently not used and may be left blank.***
94 – 250	Filler Space	PIC X (157)	

[1] This field will be required for future tracking of intraday files processed by OCC as final trades.

11 Back Office Requirements

11.1 102 Form for Regulatory

11.1.1 Assumptions

This section describes the assumptions regarding 102 Forms for this release.

11.1.1.1 General Assumptions

11.1.1.1.1 CBOT Single-Stock Future Traders Clearing through CBOT

CBOE, on behalf of OneChicago, will track, request, and store 102 Forms for SSF traders that clear through the CBOT as it relates to Single Stock Futures. Unsure whether or not CBOT or CME will also be tracking who needs to file 102 Forms for those SSF traders that clear through CBOT.

11.1.1.1.2 CME Single-Stock Future Traders Clearing through CME

CBOE, on behalf of OneChicago, will track, request, and store 102 Forms for those SSF traders that clear through the CME. CME will also be tracking who needs to file a 102 Forms for these same SSF traders that clear through CME and requesting 102 Forms from them.

11.1.1.1.3 CME Needs from CBOE

The Chicago Mercantile Exchange will not need a copy of any of the 102 Forms filed at CBOE. CME will still monitor and request 102 Forms from their members when necessary as they still need to track 102 forms for their members with large lot positions in other products in addition to single-stock futures.

11.1.1.1.4 CBOT Needs from CBOE

The Chicago Board of Trade will not need a copy of any of the 102 Forms filed at CBOE.

11.1.1.1.5 CME/CBOT Data Feeds to CBOE

CBOE will not be receiving any 102 Forms electronic or paper from either CME or CBOT.

11.1.1.1.6 What will be mailed

A blank 102 Form and a generic cover letter explaining the requirement and instructions to file the 102 Form will be mass copied and pre-stuffed into envelops ready for mailing.

11.1.2 General Requirements

11.1.2.1 Filing 102 Form Requirement

Any OneChicago member with positions in excess of 1000 or more contracts need to have filed a 102 form file with the Commodity Futures Trading Commission (CFTC).

OneChicago's Regulatory System will handle determining which individuals or firms meet this 102 Form filing requirement.

11.1.2.2 Requesting 102 Forms

CBOE's Regulatory Department, on behalf of OneChicago, will send 102 Form requests to all individuals and firms that meet the 102 Form filing requirement. Even though the CME and or CBOT may be also requesting 102 Forms to be filed, CBOE will still send their own 102 Form filing requests.

If a OneChicago member, as part of a joint account, has positions in excess of 1000 or more contracts, each participant of the joint account will be requested to complete a 102 Form.

11.1.2.3 Collecting 102 Form

CBOE, on behalf of OneChicago, will collect and store electronically the data from the 102 Forms submitted by individuals and firms that met the 102 Form filing requirement. The 102 Form will still need to be sent to the CFTC by the individual or firm and a copy sent to the CBOE. The CBOE will not receive any 102 Forms from the CME, CBOT or CFTC.

11.1.2.4 Membership's Data Needs from OneChicago's Regulatory Systems

OneChicago's Regulatory systems will communicate to Membership the individuals and firms (ideally the ENT_UIDs of the account triggering the 102 Form requirement) who should have filed a 102 Form. Membership will perform the matching between those that have filed a 102 Form and those who should file a 102 Form to determine who OneChicago's Regulatory Department needs to request a 102 Form from.

11.1.2.5 Tracking 102 Form Requests

The Membership System will track the date when an individual or firm meets the 102 Form filing requirement, the date when OneChicago's Regulatory Department mails the 102 Form request, and after the 102 Form is received, track the date a 102 Form is entered electronically into Membership.

11.1.3 Regulatory Requirements

11.1.3.1 Data Entry

A window is required to allow Regulatory to enter in the information from a 102 Form into Membership relating the 102 Form data to the person or organization that is filing the 102 Form.

This window will allow 102 Forms to be entered regardless of exchange the individual or organization is associated with.

Only Regulatory will be able to enter or modify 102 Form information. Note: SSN, name and address of the filer can only be updated by Membership, see Membership Requirement below.

When entering an associated account, a search window will be provided to search by account number and to select a previously entered account. If no match can be identified, allow regulatory to add the account number and associate it to the 102 Form filer.

When entering in an individual or organization, a search window will be provided to search by name and to select an individual or organization already defined in the Membership System. If no match can be identified, allow the regulatory user to add the individual or organization and associate them to the 102 Form filer.

There will not be any "required" fields as we do not want to prohibit the entry of incomplete data if the filer did not provide all the correct information. The only validation supplied by the window will be on the format of dates and phone numbers.

The Regulatory Department will have the ability to change any of the 102 Form information entered and at any time, except SSN, name, and address of the filer, or contact person and title. These changes must be performed by the Membership Department.

11.1.3.2 Viewing 102 Information

A facility is needed to allow Regulatory to view the entered 102 information for an individual or organization.

Regulatory also needs the ability to view a list of any special groups an individual or organization is part of. In addition, the user will also be able to select a special group in the list and open a window to display all the participants of that special group. ***(Should this be browse only or should I allow the user to also edit the special group?)***

11.1.3.3 Rectifying an Association

Regulatory needs to be able to correct 102 Form associations. For example, when entering in a person who has 10% financial interest in the account that has the name "J. Smith" on the form, but Membership only has a "Jerome Smith" and a "Jasper Smith" in Membership, the user can add "J. Smith" to the Membership system and associate him to the 102 Form due to lack of information to decide who is the correct Smith. At a later time, we learn that "J. Smith" is actually "Jerome Smith", Regulatory needs the ability to release the 102 Form link to "J. Smith" and change that to "Jerome Smith". This will not remove "J. Smith" from Membership, only the 102 Form link. Removing an entity is a manual process that at this time can only be performed by the Systems Department supporting Membership Online. But, at least the correct association can be made.

11.1.3.4 Mailing Labels

OneChicago's Regulatory System will pass, nightly, to the Membership Online a list of entities that have reached the requirement to file a 102 Form, regardless if they have or have not previously reached this threshold. The Membership system will compare this list of who should file to the list of entities that Regulatory has entered a 102 Form for to produce a list of entities that still need to file a 102 Form.

Regulatory needs the ability to view this list of entities that OneChicago needs to request 102 Forms from. This list will contain individuals or organizations that OneChicago has already sent a 102 Form request, but have not yet received a response and those that just reached the 102 Form requirement for the first time and OneChicago has not sent a 102 Form request, yet.

The user will have the option to generate mailing labels for any combination of entities in the list. The default will be only those new entities which a 102 Form has not been previously been requested.

The mailing labels will have the acronym associated with the account that triggered the need to file a 102 Form printed next to the name. Thus, a trader who is part of a joint account that triggered the 102 Form filing requirement will have the joint account placed on the mailing label and not the individual's personal acronym.

After the mailing labels are printed, the user needs to enter the date these mailing labels will be applied and mailed out. This will allow the system to track when 102 Form requests were mailed for each individual or firm meeting the 102 Form filing requirement.

11.1.3.5 Printing Requirement other than Mailing Labels

Currently, there is no requirement to print out a 102 Form with the data captured in Membership Online.

11.1.3.6 102 Form Request

The actual 102 Form request will be a blank 102 Form along with a generic cover later stating the requirement to file a 102 Form to the CFTC and a copy to OneChicago. By keeping the cover letter generic and a blank 102 Form, these can be pre-stuffed (***Assuming this is the Regulatory Department?***). All that is needed is to apply a mailing label to the pre-stuffed envelope and they are ready for mailing via the US Post Office once proper postage is applied or metered.

The Regulatory Department will create the generic cover letter with the proper wording. The Regulatory Department will also create the blank 102 Form (add SSN/TIN to the CFTC Form) that will be used for pre-stuffing envelopes.

11.1.4 Membership Requirement

If the name, address, SSN/TIN, or contact person and title on the 102 Form is different than what Membership currently has in the Membership database, the change will need to be performed by the Membership Department. Thus, the 102 Form will need to be copied and forwarded to CBOE's Membership Department to complete the address, SSN/TIN, or contact person and title change. There are multiple names on the 102 Form. Only those differences with the name in 1d. must be entered by the Membership Department.

Also, only those 102 Forms that have a different name, address, SSN/TIN, contact person and title under item 1d. on the 102 Form need to be forwarded to Membership for updating.

12 Database Data Capture Requirements

12.1 Futures Open Interest Load

12.1.1 Unique Record Identifier for Open Interest

Record Type		518	
Record Description		Open Interest	
Sequencing		1-3	Record Identifier
	*	4-4	Put/Call Indicator

	5-10	Trading Symbol
	11-16	Expiration Date
*	15-15	Strike Price Integer
*	16-19	Strike Price Decimal
	20-25	Product Type

* Will be included if coding for options on futures.
There is no intent to bring up options on futures for Security Futures products, this is just a shared record type.

12.2 Futures Open Interest Record for SBT

12.2.1 Record Layout

Record Position	Note	Field Name/Description	Size	Value/ Format
01 - 03		Record Id	03	518
04 – 04	*	Put/Call Indicator	01	‘P’ = Put ‘C’ = Call
05 - 10		Trading Symbol	06	X
11 - 16		Expiration Date	06	CCYYMM
17 - 18		Filler	02	X
19 - 23	*	Strike Price Integer	05	N
24 – 25	*	Strike Price Decimal	02	N
26 - 26		Product Type	01	‘F’ = Futures ‘X’ = Options on Futures
27 - 32	*	OPRA Class/Symbol	06	X
33 – 34	*	OPRA Code	02	X
35 – 39		Unit of Trade/Premium Mult	05	N
40 - 40		CBOE Traded	01	‘Y’ or ‘N’
41 - 41		AMEX Traded	01	‘Y’ or ‘N’
42 - 42		PHLX Traded	01	‘Y’ or ‘N’
43 - 43		PSE Traded	01	‘Y’ or ‘N’
44 - 44		Open	01	‘Y’ or ‘N’
45 - 45		NLX Traded	01	‘Y’ or ‘N’

Record Position	Note	Field Name/Description	Size	Value/ Format	Decimal
46 - 46		Open	01	'Y' or 'N'	
47 - 47		ISE Traded	01	'Y' or 'N'	
48 - 48		Dual Traded	01	'Y' or 'N'	
49 - 50		Filler	02	X	
51 - 57		Open Interest Quantity	07	N	
58 - 59		Filler	02	X	
60 - 67		Activation Date	08	CCYYMMDD	
68 - 75	*	Ex-Dividend Date	08	CCYYMMDD	
76 - 77		Filler	02	X	
78 - 83		Deliverable Symbol	06	X	
84 - 88	*	Strike Price Adjustor	05	N	3
89 - 142		Filler	54	X	
143 - 150		Current Date	08	CCYYMMDD	

**** Will be blank for 'F' product types***