CBOE Financial Network

Book Depth Dissemination

For

OneChicago Security Futures Products

Vendor Interface Specification

Ver 1.3
December 07, 2005

TABLE OF CONTENTS

Change Notice

Date	Version	Description of Change
Dec 07, 2005	V1.3	Changed section 6.11 – Sell side code changed from S to A.
May 20, 2005	V1.2.2	Corrected the price field in the Book Depth examples to have 8 bytes instead of 7 bytes.
Dec 20, 2004	V1.2.1	• Corrected the format for the Book Depth (Full Refresh, Update) message. • Changed the version number to 1.2.1 to correspond with the Market Data documents
Jan 13, 2004	V1.2.0	Added message category R in the processing logic

1.0 INTRODUCTION

OneChicago has contracted with the Chicago Board Options Exchange (CBOE) to provide dissemination services for all OneChicago (ONE) products. As supplier of the ONE match engine, CBOE is uniquely positioned to provide book depth data in the most timely fashion available. The infrastructure to support the book depth distribution will be known as the CBOE Financial Network (CFN).

This document describes the required formats for communications and output messages to effectively interface with the CFN facilities.

2.0 GENERAL DESIGN OF DATA DISTRIBUTION NETWORK

CFN will utilize a TCP/IP multicast protocol for the dissemination of ONE information. CFN transmits variable length blocks having maximum lengths of 1000 characters. The length includes a Block Length Header, text and control characters. The number of messages contained in a block is variable and the end of each message is delimited by a Unit Separator character (US), except for the last record in the block, which ends with an End of Text (ETX) character.

Data available via the CFN IP multicast distribution network includes:

Real-Time Production Data
A copy of each ONE real-time production message is available from each connection to the CFN facility delivered via two distinct multicast data streams.

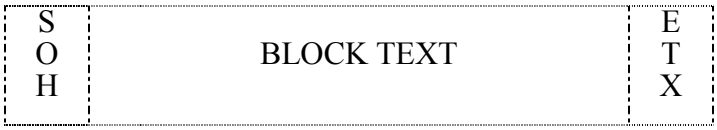
Pre-Launch Test Data
Prior to the official launch of OneChicago markets test data will be disseminated over CFN production ports. After launch, OneChicago will approve new vendors for interim connections to the production feed from vendor test environments until they are prepared to move to their production environment.

Time Beacon and Retransmission facility
CFN does not anticipate implementing either of these facilities initially. The evolution of market data networks over the past several years has dramatically reduced the number of retransmission requests made by vendors. As the network and user community of CFN expands a needs analysis will be performed to determine the importance of providing these features.

3.0 TRANSMISSION CHARACTERISTICS

3.01 TRANSMISSION BLOCK

Encapsulated within each IP packet is a transmission block. One type of transmission block is used for all types of messages:



|<------ TRANSMISSION BLOCK ------->|

A block can have a maximum of 1,000 characters inclusive of text.

3.02 SOH AND ETX

The Start of Header (SOH) control character (x01) indicates the beginning of the block, whereas an End of Text (ETX) control character (x03) signifies the end of the block.

3.03 US

The Unit Separator (US) control character (x1F) is needed in multiple message blocks to signify the end of the preceding message but not the end of the block.

3.04 BLOCK TEXT

The block text can consist of multiple messages. A message is a unit of data that can be processed by the receiving station independently of other data. A message may not span a block boundary.

A message consists of a Message Header, which is of fixed length and format, and a Message Text segment that is variable in length and format. A US character delimits each message, while an ETX character delimits the last message in the block.

3.05 BLOCK TEXT FORMAT

The block text consists of multiple messages with each message consisting of a Message Header and, with the exception of certain control messages, message text.

The block text is depicted below:

S O H	MESSAGE 1 HEADER & TEXT	U S	MESSAGE 2 HEADER & TEXT	U S	~ ~ ~ ~	MESSAGE n HEADER & TEXT	E T X

|<---------- BLOCK TEXT ---------->|

3.06 DATA FORMAT

ASCII filler characters are inserted, as required, in accordance with the following rules:

1. Zeros (hex 30) are inserted in Numeric fields. All Numeric fields are right justified, as required.

2. Spaces (hex 20) are inserted in Alphabetic fields. All Alphabetic fields are left justified, as required.

3. Spaces (hex 20) are inserted in Alphanumeric fields, except Strike and Premium Price Denominator Code fields, which are Zero filled. All Alphanumeric fields are right or left justified, as required.

4. Reserved fields will be space filled.

3.07 CHARACTER SET

All transmissions are in standard 8-bit ASCII code.

4.0 MESSAGE HEADER

The Message Header supplied on each message contains a total of **18 Bytes** and conforms in all cases to the following data fields:

EXCHANGE ID	Reserved	MESSAGE IDENTIFICATION	MESSAGE SEQUENCE NUMBER	TIME
1	**1**	**2**	**8**	**6**

|←---------- 18 BYTES ----------→|

4.01 MESSAGE HEADER FIELD DESCRIPTIONS

4.02 EXCHANGE ID

The Exchange ID field is a **1 Byte**, Alphabetic character that identifies the ONE as the initiator of the message:

CODE	**VALUE**
E	OneChicago, LLC

4.03 RESERVED

The Reserved field is a **1 Byte**, Alphanumeric character, ASCII space filled field. Field is reserved for future use.

4.04 MESSAGE IDENTIFICATION

The Message Identification field is a **2 Byte**, Alphabetic upper or lower case character, Space filled field. The first character entered in the Message Identification field identifies the **Message Category**. The second character entered in the Message Identification field identifies the **Message Type**. The Message Type character is Space filled to either indicate a specific value, or that a Message Type is not applicable to a specified Message Category.

LOWER CASE CODE	**VALUE**
t	**Book Depth Full Refresh**
u	**Book Depth Update**

4.05 MESSAGE SEQUENCE NUMBER (MSN)

The Message Sequence Number (MSN) is an **8 Byte**, Numeric, Right Justified, Zero filled field. All messages are assigned a sequence number. On a per line basis, the MSN on the lines are set to Zero at the start of each day and are incremented by one each time a message (other than a Retransmission or Reset Message Sequence Number message) is transmitted. The following exceptions or special considerations should be noted:

1. Retransmitted messages contain the MSN of the original message.

Should CFN experience a line failure and recovery, the message sequence number for the recovered line(s) is reset to a sequence number greater than the last number transmitted. A sequence reset message will be transmitted prior to resumption of data transmission and will indicate the new sequence number that the receiver should expect.

4.06 TIME

The Time (Central Time) field is a **6 Byte**, Numeric Character, Zero filled field. Format is (Military) HHMMSS where HH=Hour, MM=Minute and SS=Second. Indicates the Time that a transaction is disseminated. For Retransmissions, the Time will contain the original Time that the transaction was disseminated.

5.0 MESSAGE FORMATS

Each message transmitted by CFN consists of a Message Header and Message Text. The particular Message Category and Message Type entered in the Message Identification field of the Message Header determines the format of the text. Message formats are fixed field formats. Detailed information on each field specified in every message format is contained in alphabetical order in the **Field Descriptions** section **(section 6.0)** of this document.

5.01 BOOK DEPTH (Full Refresh, Update)

Variable Length 25 + (17 * *NUM_ENTRIES*) BYTES

The Book Depth Full Refresh and Book Depth Update messages use the same message format. The messages support efficient transmission of changes to book depth up to a system specified depth - currently set at the top five bid and asks for a product. The bid and ask price entries are the composite for all quotes and orders resting against a product.

CATEGORY	TYPE
t,u	**Space filled**

SYMBOL	RESERVED	EXPIRATION DATE	YEAR
6	**1**	**1**	**1**

|←---------- 9 BYTES ----------→|

RESERVED	PRICE DENOMINATOR CODE
9	**1**

|←---------- 10 BYTES ----------→|

SESSION INDICATOR	RESERVED
1	**1**

|←---------- 2 BYTES ----------→|

BOOK SIZE	NUM ENTRIES
2	**2**

|←---------- 4 BYTES ----------→|

ACTION	BOOK POSITION	SIDE	PRICE	VOLUME IN BOOK
1	**2**	**1**	**8**	**5**

|←---------- 17 * *NUM_ENTRIES* BYTES ----------→|

5.02 Book Depth Processing Logic

OneChicago Book Depth data is distributed using an incremental update approach where the entire book depth for a product is sent periodically as a refresh, starting at the beginning of trading (pre-open). Between full refreshes, updates to individual book depth entries are sent.

The Book Depth Full Refresh Message (Message Category = t) contains (2 * BOOK SIZE) book depth entries. The BOOK SIZE top bid prices and the BOOK SIZE top ask prices are sent with an action of "A' - to add. When a full refresh message is received - the buy side and ask side of the book for the product should be deleted, being replaced with the values contained in the full refresh message.

R	Replace	Replace the entry at the corresponding price level

The Book Depth Update Message (Message Category = u) contains 1 to 10 book depth entries. Each book depth entry has an Action associated with it.

A	Add	Add an entry to the buy side or sell side - ordered by price level
U	Update	Update the quantity for a price level
D	Delete	Delete the entry at the corresponding price level
R	Replace	Replace the price at a specified level in the book (implemented to eliminate need to send Add + Delete for a single entry change)

When a user listens on the multicast group they should wait until the first full refresh message per product to start processing. Once the first full refresh is received, then the application should begin to process updates for that product.

Messages are synchronized using a sequence number. The sequence number applies across products - all messages must be consumed for proper sequence number handling to keep the view of the book synchronized between full refreshes.

Example 1 First Full Refresh message of the day

t.IBM...V3.....B...0510
R01B00000100 00010
R02B00000095 00020
R03B00000090 00010
R04B00000085 00020
R05B00000075 00020
R01S00000125 00015
R02S00000130 00005
R03S00000140 00005
R04S00000155 00010
R05S00000200 00010

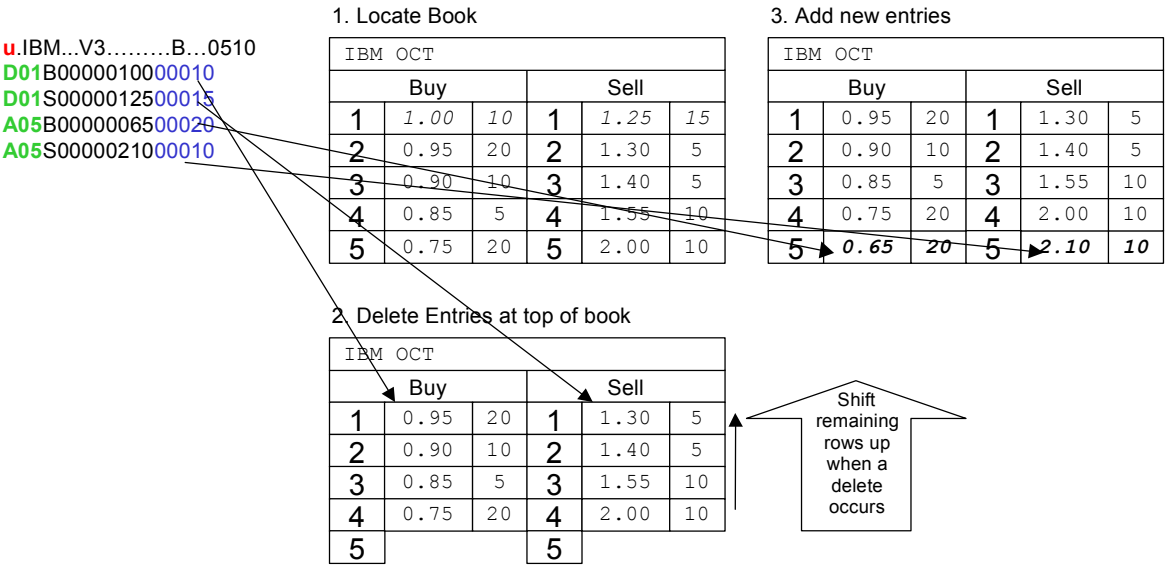
1. Create Book

Buy	Sell

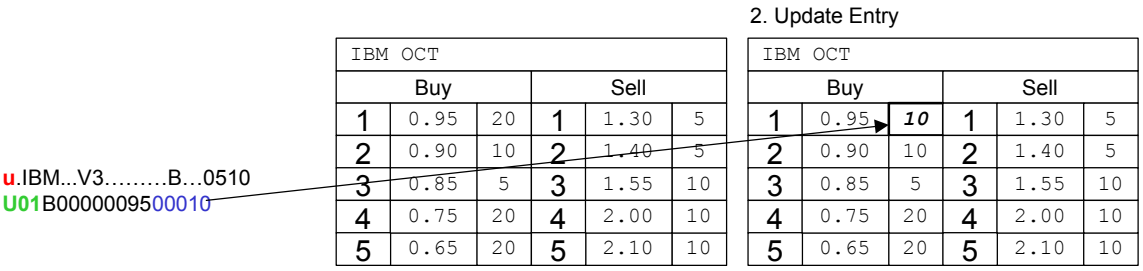
2. Add Buy Side and Sell Side to Book

IBM OCT					
Buy			Sell		
1	1.00	10	1	1.25	15
2	0.95	20	2	1.30	5
3	0.90	10	3	1.40	5
4	0.85	5	4	1.55	10
5	0.75	20	5	2.00	10

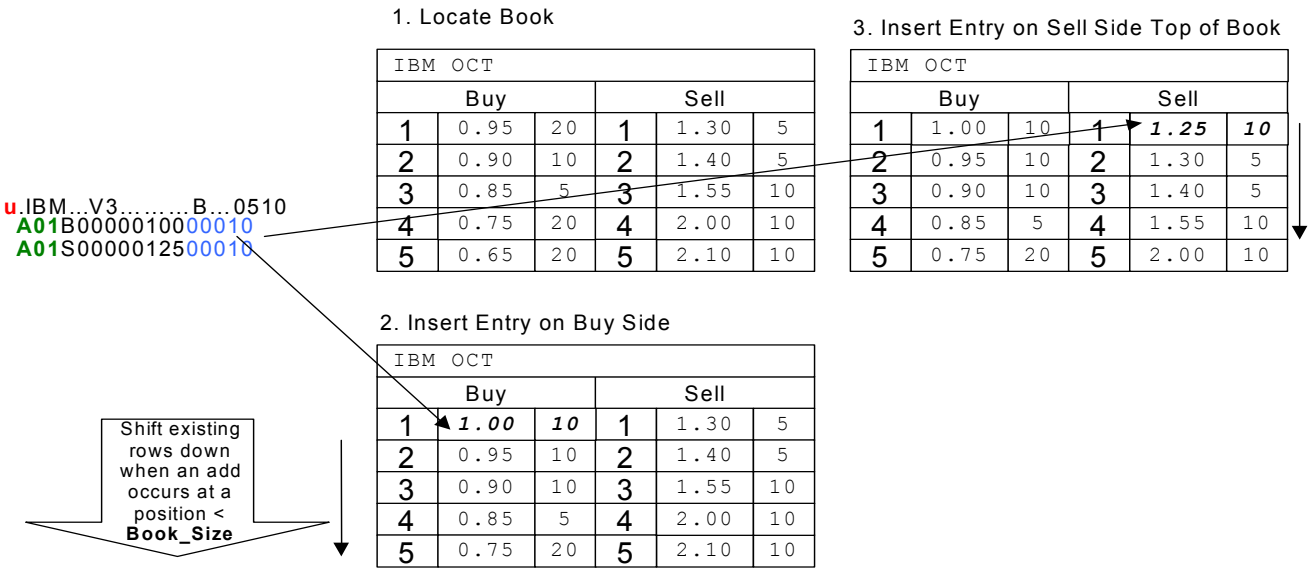
Example 2 Top of the book is traded



Example 3 Top of the book partially traded



Example 4 New Top of the book entry on both sides



IBM OCT

	Buy			Sell	
1	0.95	20	1	1.30	5
2	0.90	10	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	0.65	20	5	2.10	10

IBM OCT

	Buy			Sell	
1	1.00	10	1	1.30	5
2	0.95	10	2	1.40	5
3	0.90	10	3	1.55	10
4	0.85	5	4	2.00	10
5	0.75	20	5	2.10	10

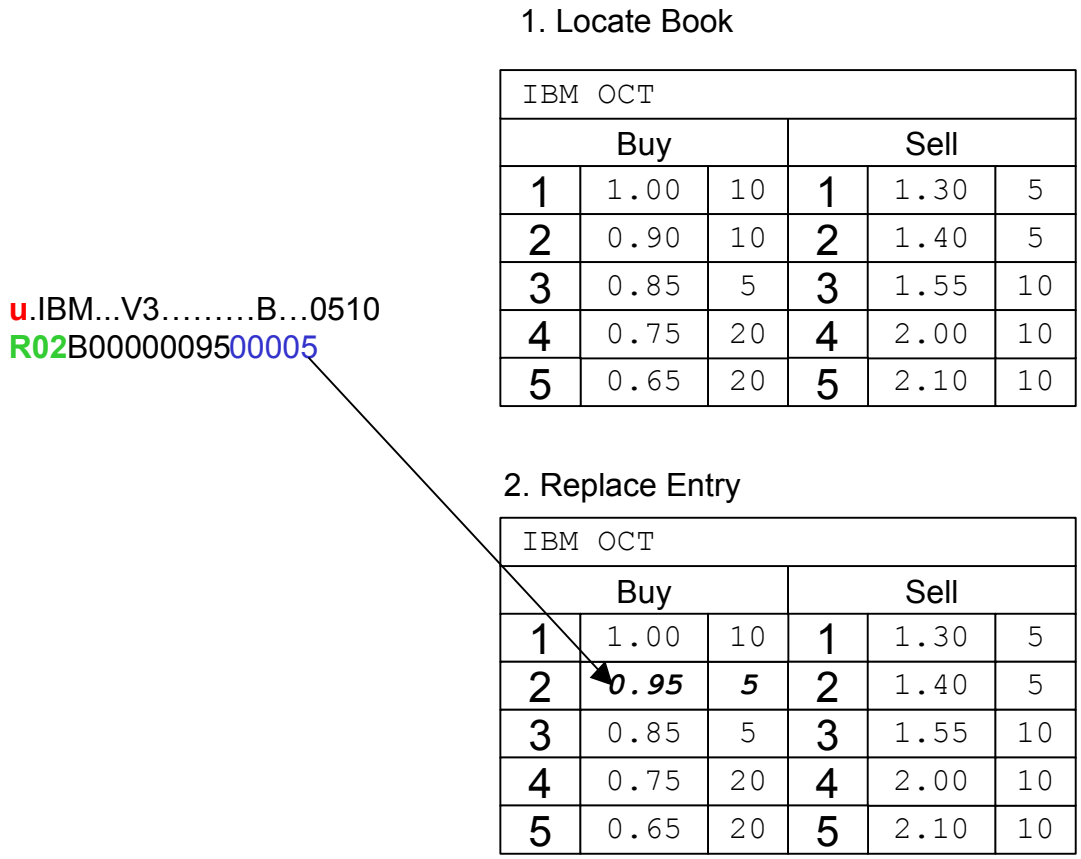
IBM OCT

	Buy			Sell	
1	1.00	10	1	1.25	10
2	0.95	10	2	1.30	5
3	0.90	10	3	1.40	5
4	0.85	5	4	1.55	10
5	0.75	20	5	2.00	10

IBM OCT

	Buy			Sell	
1	1.00	10	1	1.30	5
2	0.90	10	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	0.65	20	5	2.10	10

Example 5 Replace entry in book



IBM OCT

	Buy			Sell	
1	1.00	10	1	1.30	5
2	0.90	10	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	0.65	20	5	2.10	10

IBM OCT

	Buy			Sell	
1	1.00	10	1	1.30	5
2	0.95	5	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	0.65	20	5	2.10	10

Example 6 Replace last entry in book

u.IBM...V3……B…0510
R05B0000007000025

1. Locate Book

IBM OCT					
Buy			Sell		
1	1.00	10	1	1.30	5
2	0.95	5	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	0.65	20	5	2.10	10

2. Replace Entry

IBM OCT					
Buy			Sell		
1	1.00	10	1	1.30	5
2	0.95	5	2	1.40	5
3	0.85	5	3	1.55	10
4	0.75	20	4	2.00	10
5	***0.70***	***25***	5	2.10	10

Network Connectivity

OneChicago Book Depth data is distributed using separate CFN multicast groups. The service is designed to be stateless in operation. There is no retransmission capability provided for book depth. There are no control messages on the book depth multicast groups.
Products are distributed alphabetically across multiple multicast groups.
A-E - Multicast Group One
F-N - Multicast Group Two
O-Z - Multicast Group Three

6.0 FIELD DESCRIPTIONS

ASCII code characters are defined as follows:

TERMINOLOGY	DESCRIPTION
Alphabetic	ASCII characters "A" – "Z" or "a" – "z"
Numeric	ASCII characters "0" – "9"
Alphanumeric	Any combination of Alphabetic and Numeric as defined above
Special and Punctuation	Any printable ASCII character **except** Alphanumeric as defined above, i.e.; White Space

6.01 ACTION

1 Byte, Alphabetic. Indicates what action is to be applied with the book entry. Values are:

CODE	ACTION	DESCRIPTION
A	Add Entry	New entry to be added at a position - Subsequent positions are moved down 1 position
U	Update Entry	Update the bid or ask size at a book position
D	Delete Entry	Delete an entry at a location. Subsequent positions are moved up 1 position.
R	Replace Entry	Replace values at a book entry with values provided

6.02 BOOK POSITION

2 Bytes, Numeric. Right Justified, Zero filled.

The Book POSITION specifies where in the book an update action is to take place.

6.03 BOOK SIZE

2 Bytes, Numeric. Right Justified, Zero filled.

The Book Size specified the maximum number of book positions (price levels) are being reported. The Book Size is configurable. It is important to be sure your application can handle changes in size of the book.

6.04 EXPIRATION DATE

1 Byte, Alphabetic. It indicates the expiration month for the security future.

CODE	VALUE
F	JAN
G	FEB
H	MAR
J	APR
K	MAY
M	JUN
N	JUL
Q	AUG
U	SEP
V	OCT
X	NOV
Z	DEC

6.05 NUM ENTRIES

2 Bytes, Numeric. Values 00-99. The number of book depth entries contained in the message.

6.06 PRICE

8 Bytes, Numeric. Right Justified, Zero filled. When there is no Price, this field is Zero filled.

The whole and decimal portion of Price information. The Price Denominator Code determines the location of the decimal point.

Represents the price of a contract, determined in the competitive marketplace, which the buyer pays to the seller for the rights conveyed by the contract.

6.07 PRICE DENOMINATOR CODE

1 Byte, Alphanumeric. The Price Denominator Code field indicates the position of the floating decimal point.

Price Denominator Code	Denominator Value
A	10
B	100
C	1,000
D	10,000
E	100,000
F	1,000,000
G	10,000,000
I	(N/A)
H (not defined)	

6.08 RESERVED

Variable (V) Bytes, Alphanumeric. Reserved fields are Space filled unless otherwise defined.

Fields reserved for future use.

Note: **Reserved fields will be redefined and implemented as required. If data recipients are not prepared to process new field(s) when implemented, they should be able to handle them to the extent that they do not impact their normal data processing.**

6.09 SECURITY SYMBOL

6 Bytes, Alphabetic. Left Justified, Space filled.

Identifies the unique symbol assigned to the security.

6.10 SESSION INDICATOR

1 Byte, Alphabetic, Space filled.

The Session Indicator identifies the session in which the trade or quote originated.

CODE	VALUE
a (lower case)	Morning (A.M.) session
Space filled	Normal session

6.11 SIDE

1 Byte, Alphabetic, The side of the book represented by the book entry.

CODE	VALUE
B	Buy side of book position
A	Sell side of book position

6.12 VOLUME IN BOOK

5 Bytes, Numeric. Right Justified, Zero filled. When there is no PRICE, this field is Zero filled.

The VOLUME identifies the number of contracts associated at a given price within the book for a contract. The SIDE field determines if the entry is a buy or sell.

6.13 YEAR

1 Byte, Alphanumeric, Space filled.

Contains the last digit of the expiration year.