TMX Information Processor

Canadian Best Bid and Offer (CBBO®)

Canadian Best Bid and Offer - Protected Only (CBBOP®)

Functional Specifications

Version 2.0

Published: June 4, 2015

Effective: September 21, 2015

Revision History

Version	Date	Changes
2.0	June 4, 2015	Section 2.2 – Added ServiceID example “CP1” identifying CBBO Protected Only Section 5.2 – Updated TSX Alpha Exchange Open/Close times Section 5.2 – Removed references to TMX Select Section 6.0 - Removed references to Alpha Intra-Spread (AIS) and TMX Select Section 6.0 - Updated NumericPrice to reflect precision of up to 5 decimal places ([“.” 1*5Digit])
1.9	December 19, 2014	Section 5.2 – Added Aequitas (Lit , Neo and Crossing Facility) marketplace hours Section 6.0- Revised ExchangeID (tag 247) to include Aequitas
1.8	October 3, 2014	Section 3.1 - Removed TimeStamp and LasTSXquenceReceived tags. Changed CdfPubTimeStamp and CdfRcvTimeStamp from mandatory to optional tags. Changed SourceAddress from optional to mandatory tag Section 5.2 – Updated Open time for Omega and Lynx Section 7.0 – Removed LastMessage (Tag 113), NonResident (Tag 168) and Price (Tag 41) Section 8.0 – Removed not applicable Tags from Field Identifiers (Numerical) Section 9.0 – Removed not applicable Tags from Field Identifiers (Alphabetical)
1.7	July 31, 2014	Section 5.1 (4) – Updated Start of Day quotation time from 3:25 to 6:10 AM.
1.6	July 30, 2014	Section 5.2 – Removed reference to PURE Trading and updated reference to CSE Section 7 – Updated description of CSE ExchangeID (Tag 247)
1.5	November 13, 2013	Updated 5.2 – Operational Hours ExchangeId added “LYX” for Lynx ATS
1.4	February 11, 2013	Updated 5.2 – Operational Hours ExchangeId added “CTH” for CX2
1.3	November 29, 2011	Removed Field [TimeStamp] from Quote message
1.2	November 2, 2011	Format updated Updated section 5.1 to include information about start of day messages Updated section 5.2 to provide clarity on the CBBO’s dessimination of data during each market places regular trading session.
1.1	March 31, 2011	Revised Section 5.2 Trading Hours for the CBBO to include TMX Select
1.0	February 09, 2011	Revise Tag 247 ExchangeId to include “SEL” for TMX Select
0.4	January 26, 2010	Exchange Identifier and Exchange Admin Added “A” for Alpha ATS ExchangeId added “ALP” for Alpha ATS Changed references to CDF CBBO to TMX IP CBBO Updated Disclaimer

0.3	January 05, 2009	Added the following internal tags to identify messages in the consolidation process. Added CDFId to Control Header Content Added ConsInboundTimeStamp to Control Header Content Added ConsOutboundTimeStamp to Control Header Content
0.2	October 13, 2008	CBBO Exchange Identifier ("Q")
0.1	October 7, 2008	CBBO Service Id and Message Type

NOTICE & DISCLAIMER

The information contained in this document and any other information or documentation related to the CBBO specification including all past, present and future versions thereof and associated message formats (collectively referred to as the "CBBO Specification") are provided "as is", and neither TSX nor any of its affiliates makes any representation or warranty, express or implied, as to the contents of the CBBO Specification (or the results obtained by using it) and each such person specifically disclaims any representation, warranty or condition of accuracy, completeness, merchantability, fitness for a particular purpose or that it is error-free. By using the CBBO Specification, you are agreeing to assume the entire risk associated with its use.

TSX and its affiliates shall have no liability for damages of any kind arising in any manner out of or in connection with your use of, or your inability to use, the CBBO Specification, whether direct, indirect, incidental, special or consequential (including, without limitation, loss of data, loss of use, third party claims, lost profits, lost revenues and strict liability), arising under contract or otherwise, whether or not TSX has been advised of, or otherwise might have anticipated the possibility of, such damages.

The information contained in the CBBO Specification is proprietary and confidential information belonging to TSX and its licensors or service providers, as applicable. Copyright and trade-mark rights and any other intellectual property rights in the CBBO Specification belong to TSX (or its licensors or service providers, as applicable). Your permitted use of the CBBO Specification, in whole or in part, is limited to the non-exclusive, non-transferable, revocable, non-assignable, personal right for you only to build a connection between your systems and TSX's market information system. Any uses not specifically set out herein are prohibited. TSX may revoke this right at any time, on notice to you, and you agree to promptly comply with such notice. Do not reproduce, disclose, distribute, publish, sell, modify, commercialize or make any part of the CBBO Specification available, in whole or in part, to any person including to any of your affiliates, service providers or any other related party without TSX's prior written consent. If you do not agree with, or do not wish to comply with, any of the above, please notify TSX and, in accordance with TSX's instructions, either return or destroy any copies or version of the CBBO Specification in your possession.

This Notice, Disclaimer and Rights Granted sets out the terms and conditions on which you are permitted to use the CBBO Specification. This Notice, Disclaimer and Rights Granted may be changed from time to time by TSX in its discretion. It is your obligation to check and comply at all times with the terms applicable to the most current version of the CBBO Specification.

Table of Contents

1. Overview

This document describes the Canadian Best Bid and Offer (CBBO) data feed provided by the TMX IP.

The CBBO is a multi-market Best Bid and Offer pricing feed that will allow clients to price orders at or between the best bid and best offer among multiple market centres and to provide basic information for routing of orders.

The CBBO business content messages are formatted using the STAMP protocol syntax. STAMP, the Securities Trading Access Message Protocol, is the messaging protocol developed by TSX for order entry. More details about the STAMP protocol are given in the *STAMP Specification* (Reference [1]).

1.1. Intended Audience

The intended audience of this specification are business analysts and programmer analysts.

All readers should familiarize themselves with *Section 1.2* – Rule Notation Conventions, paying close attention to how the notation conventions are defined, as this notation is used throughout the specification.

Business analysts should focus primarily on Section 4, Business Content Messages and Section 8, Data Dictionary. These two sections define how the trading information is defined in the CBBO. In addition to these sections, the business analysts should be familiar with the trading rules and trading scenarios that these messages represent.

Programmer analysts should be familiar with the entire specification, although their focus should be on message structure and parsing.

1.2. Rule Notation Conventions

This section describes the notation convention for the elements of STAMP syntax used in the business content. Although the rules presented below are somewhat formal in nature, for casual reading of the specification all that is required is to keep in mind the following points:

- Text presented in a `typewriter typeface` font means that it is a rule that is defined in the Data Dictionary starting on page 14.
- Any rule that is enclosed in square brackets, "[" and "]" means that the rule is optional.
- The spaces between the rules means that the rules are joined together.

When appropriate, this specification uses an augmented Backus-Naur Form (BNF) notation, similar to that presented in *RFC 822 – Standard For The Format of ARPA Internet Text Messages* (Reference [8]). The differences from standard BNF involve naming rules and indicating repetition and "local" alternatives. Comments about a rule, such as the hexadecimal representation of a character, are introduced by a semicolon (";") in-line after the rule definition. All text after a semicolon until the end of a line forms the comment.

Rules are used throughout the text of the specification when appropriate to formally define a concept. All of the rules are gathered in the Data Dictionary on page 14 for convenience.

1.2.1. Rule Naming

Angle brackets ("<", ">") are used below in the syntax definition of rules to identify rule components; these brackets are not used, in general, in the rule names. The name of a rule is simply the name iTSXlf, rather than "<name>". Capitalized letters are used in names to highlight the meaning of the name.

1.2.2. Literal Text

Quotation marks enclose literal text (which is case sensitive). Literal text appears as is in the message content.

1.2.3. Alternatives: Rule1 | Rule2

Elements separated by vertical line (“|”) are alternatives. Therefore, “[abc | def]” will accept abc or def.

1.2.4. Local Alternatives: (Rule1 | Rule2)

Elements enclosed in parentheses are treated as a single element. Thus, “(elem (abc | def) elem)” allows the token sequences “elem abc elem” and “elem def elem”.

1.2.5. Repetition: *Rule

The character “*” preceding an element indicates repetition. The full form is:

```
<l>*<m>element
```

indicating at least <l> and at most <m> occurrences of element, with default values of 0 and infinity respectively.

So that “*(element)” allows any number, *including zero*; “1*element” requires at least one; and “1*2element” allows one or two.

If the repeated element is a FieldIdentifier, the repeated element will be represented in the datastream using the FieldIdentifierIndex notation as described in *Section 2 of the STAMP Specification Version 3.0*.

1.2.6. Optional: [Rule]

Square brackets enclose optional elements; eg., “[abc def]” is equivalent to “1*1(abc def)”. The square bracket notation is used in the message description.

1.2.7. Specific Repetition: Nrule

“<n>(element)” is equivalent to “<n>*<n>(element)”; that is, exactly <n> occurrences of (element). Thus 2Digit is a 2-digit number, and 3AlphaNumeric is a string of three alphabetic characters. If the repeated element is a STAMP FieldIdentifier, the repeated element will be represented in the datastream using the FieldIdentifierIndex notation as described in *Section 2 of the STAMP Specification Version 4.0.*

1.2.8. Client/Server Notation Convention

For the purpose of this specification, “Client” (initial capital letter) refers to the computer application that “listens” for output messages from the CDF CBBO service.

2. Service Architecture

The CBBO service adhere to TSX service architecture for market data dissemination defined in reference [3].

2.1. Framing

CDF market data message uses the following basic structure:

STX	Transport Header	Message	ETX

where, STX is the Start of Text (Hexadecimal 0x02), and ETX is the End of Text (Hexadecimal 0x03), "Message" is the business content that is described in Section 3 and 4.

2.2. Transport Header

The "Transport Header" is a is a 22-byte section coded in ASCII and structured as follows:

Field	**Length**	**Contents / Values**
Length	4	Total length of header and message business content (excludes STX and ETX), padded with zeros to the left.
Sequence Number	9	Sequence number assigned at service broadcast, padded with zeros to the left. Blank on Heartbeat messages.
ServiceID	3	"CB1" Code identifying the service CBBO service "CP1" Code identifying the Protected CBBO service.
Retransmission Identifier	1	0 – Normal transmission 1 – Message being sent out of order from their generation by the trading system. This can be due to unusual processing causing delay or recovery from a problem or link failure.
Continuation Indicator	1	0 – This is stand alone packet (the message fits in one packet) 1 – This packet continues in the next packet (the message spans at least 2 packets). 2 – This packet is the continuation of the previous packet. 3 – This packet is both the continuation of the previous packet and continues in the next packet.
Message Type	2	"V" for Heartbeat message (padded with a blank to the right). Left blank for all other message types.
Exchange Identifier	2	Code assigned to the originating exchange (padded with a blank to the right) as follows: "Q" for CBBO Quote

Every message packet is assigned a sequence number from 000000001 to 999999999 (decimal ASCII), with wrap-around. The sequence is reset to 1 each day and it is incremented by 1 for each packet sent.

2.3. Heartbeat Message

The Heartbeat message is sent every 60 seconds and is unsequenced. The Heartbeat message provides three information sections regarding real time message delivery, delimited by brackets:

- ◊ HEARTBEAT section, including date and time and decimals seconds since 1970 up to the microsecond,
- ◊ LAST SENT section, including sequence number of last message sent, time sent, and decimal seconds up to the microsecond,
- ◊ LAST HB section, including the “last sent” information passed in the last heartbeat message sent.

The information provided in the Heartbeat message allows clients to track real time delivery latencies.

The Heartbeat message is a fixed field length message with the following format:

Field	Length	Value /Definition	Description / Format
	1	“[“	Separator
	10	“HEARTBEAT”	Section identifier
Date	10		Date in format YYYY-MM-DD
	1	blank	Separator string
TimeOfDay	8		Time of day in format HH:MM:SS
	1	“-“	Separator string
SecondsSince1970	19	6 decimals with embedded decimal point	Formatted with “%012d.%06d” in C language
	2	“][	Separator
	10	“LAST SENT “	Section identifier
SeqNbrOfLastMsgSent	9		Last sequence number sent, padded with 0s to the left
	1	“-“	Separator
TimeLastMsgSent	8		Time last message sent in format HH:MM:SS
	1	“-“	Separator
SecondsSince1970LastMsg	19	6 decimals with embedded decimal point	Formatted with “%012d.%06d” in C language
	2	“][“	Separator
	10	“LAST HB “	Section identifier – Last Heartbeat data, right-padded with blanks.
SeqNbrOfLastMsgSent	9		This number lets the client know if they missed a heartbeat
	1	“-“	Separator
TimeLastMsgSent	8		Time last message sent in format HH:MM:SS in last heartbeat
	1	“-“	Separator
SecondsSince1970LastMsg	19	6 decimals with embedded decimal point	Formatted with “%012d.%06d” in C language in last heartbeat
	1	“]”	Separator
OCSAsubject	20		TSX diagnostics
OCSAinstance	2		TSX diagnostics
Hostname	8		ID of the originating host.
Version	4		Version of the service being delivered

The following is an example of a heartbeat message:

```
^B0207         CB100V Q [HEARTBEAT 2013-01-07 04:00:03-
001357549203.104923][LAST SENT 000019267-03:25:21-001357547121.874938][LAST
HB   000019267-03:59:03-001357549143.105259]OCSA-CDF-1           ATDOTDR
00.1^C
```

The \02 and \03 strings represent respectively the STX and ETX characters framing the message.

2.4. Message Retransmission

CBBO will provide support for automated retransmissions as defined in reference [3].

3. Message Structure

Business content in CDF messages is coded in STAMP format. This portion of the message is formally described as follows:

MessageContent	=	ControlHeader BusinessContent [ControlTrailer]
ControlHeader	=	ControlHeaderChar ControlHeaderContent
ControlHeaderContent	=	1*ControlHeaderField
ControlHeaderChar	=	<US-ASCII SOH> ; 0x01 Start of Heading
BusinessContent	=	BusinessContentChar 1*BusinessContentField
BusinessContentChar	=	<US-ASCII FS> ; 0x1c File Separator
ControlTrailer	=	ControlTrailerChar
ControlTrailerChar	=	<US-ASCII GS> ; 0x1d Group Separator

3.1. Control Header Content

ControlHeaderContent	=	[CdfPubTimeStamp] [CdfRcvTimeStamp] DestAddress SequenceNumber SourceAddress [Retrans] [RetransId] [CdfId] [CdfInboundTimeStamp] [CdfOutboundTimeStamp]

The CDF service includes the STAMP control layer header and trailer. The STAMP control header is described in detail in reference [1]. The only STAMP header field that provides useful information in the context of service is TimeStamp.

The ControlHeaderChar (0x01), BusinessContentChar (0x1c), and ControlTrailerChar (0x1d), separators are not explicitly mentioned in Section 4, Business Content Messages.

3.2. Business Content Fields

Both the Control and Business Content Sections are further divided into *Fields*. Each field is made up of a field identifier and an optional field value. The identifiers and values are variable in length and content; the Data Dictionary must be consulted for appropriate qualifying rules.

A field is divided into two sections; a field identifier and an optional field value. The FieldIdentifier is introduced by a FieldIdentifierChar. The optional FieldValue is introduced by the US-ASCII equals sign “=”. Note that it is possible to have a FieldIdentifier without a FieldValue, in which case the FieldValue assumes a default value (see the Data Dictionary).

The formal notation for a field is:

BusinessContentField	=	FieldChar FieldIdentifier ‵‵=′′ [FieldValue]
FieldChar	=	<US-ASCII RS; Record Separator> ; 0x1e

NOTE:

The FieldIdentifier and FieldValue listed in the Data Dictionary are for reference only. Some of these fields are defined as part of the STAMP protocol but will never appear in the business content messages delivered with the CDF service.

3.2.1. *Field Ordering*

The order of the fields within a section of a STAMP message are position independent. They must only be of the correct *type* (e.g. the fields within the ControlHeader must be of the type ControlHeaderField), and may be in any order within the section.

3.2.2. *Field Identifier*

The *Field Identifier* is a number that is used as an index into the Data Dictionary in 14 to identify the syntactic meaning of the field value. As an example, if the field identifier of a field was “55”, this would mean the field value was a stock symbol.

For repeating groups of field identifiers, a “dot” notation is used. If a message contains multiple occurrences of a field identifier, each occurrence is represented by an addition field identifier index. If there are linked groups of fields the index is used to link the elements syntactically. For example, an OrderBookMessage may contain multiple fields in a message, such as “64.0=1000”, “197.0=Sell”, “41.0=13.75”, “55.0=SHK”, referring to an open sell order for symbol SHK for 1000 shares at $13.75. The tag interpretations are: tag 197 represents MarketSide, tag 55 represents Symbol, tag 64 represents Volume, and tag 41 represents Price.

It is important to note that field indexes start at zero and are contiguous. Also, a field identifier without an explicit index is equivalent to an index of zero. Fields at the same index level are conceptually “records”.

Note that the contiguous nature of the index refers to the conceptual record not individual FieldIdentifiers. For example, a STAMP message with the following tags, “11.0=ABC”,”11.1=DEF”,”15.1=5”, would be valid and would represent a situation where tag 15 was optional and not present for the “0” record. There would, however, be at least one field at each index level.

The formal notation for a field identifier is:

FieldIdentifier	=	1*4Digit [FieldIdentifierIndex]; 1 to 9999, no default

FieldIdentifierIndex	=	``.’’ 1*4Digit ; 0 to 9999, default is 0

3.2.3. *Field Value*

The *Field Value* contains the value of the field. To use a previous example, if the identifier was “55” and the value was “BCE”, then the stock symbol for this message would be “BCE”.

The formal notation for a field value is:

FieldValue	=	1*PrintableChar

4. Business Content Messages

The messages described in this section are the trading messages that are broadcast from TSX to the Client.

4.1. CBBO Quote Message

The CBBO Quote Message includes quote details including the attributed Exchange based on time priority of orders at the best bid and best offer, best bid and offer price, aggregate volumes at best bid and offer.

CBBOQuoteMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress [Retrans] [RetransId] [CdfId] [ConsInboundTimeStamp] [ConsOutboundTimeStamp]
BusinessContent	=	BusinessClass BusinessAction 2PublicPrice Symbol 2Volume 2[ExchangeId]
Where:		
BusinessClass	=	``Quote’’
BusinessAction	=	``Quote’’

Each CBBO consists of two quotes. By convention, the first element of any two element field (.0) will refer to the buy side and the second element (.1) will refer to the sell side.

5. Operating Sequence

5.1. Transmission Times

(1) Clients can listen on the CBBO port at any time during the day. The unsequenced Heartbeat message is transmitted every 60 seconds.

(2) Re-transmission requests can be sent from 5:00AM to 22:00PM.

(3) Transmission times for CBBO are Eastern Standard/Daylight Savings Time.

(4) CBBO will publish start of day quotations following system initialization, at approx. 6:10 AM

5.2. Operational Hours for the CBBO

The CBBO will disseminate market data from each market place during their regular trading session.

Exchange	**Regular Trading Session (EST)**	
	Open	**Close**
Aequitas - Lit	09 :30	16 :00
Aequitas - Neo	08 :00	17 :00
Aequitas – Crossing Facility	08 :00	17 :00
TSX Alpha Exchange *	08 :00	17 :00
Chi-X Canada	08 :30	17 :00
CX2	08 :30	17 :00
CSE Listed Securites (formerly known as CNSX)	09 :00	16 :00
CSE Other Listed Securities (formerly known as PURE Trading)	08 :00	17 :00
Omega ATS	08 :00	17 :00
Lynx ATS	08 :00	17 :00
TSX	09 :30	16 :00
TSX Venture Exchange	09 :30	16 :00

*Note – Unprotected Marketplaces are only available within the CBBO, not the CBBO for Protected Only Markets (CBBO-P).

6. References

[1] STAMP Specification

[2] Toronto Broadcast Feed Specification

[3] CDF, *Protocol Specifications and Service Access*

[4] RFC 768, *User Datagram Protocol*, J. Postel, September 1981, http://www.internet.nic

[5] RFC 791, *Internet Protocol*, J. Postel, September 1981, http://www.internet.nic

[6] RFC 792, *Internet Control Message Protocol*, J. Postel, September 1981, http://www.internet.nic

[7] RFC 793, *Transmission Control Protocol*, J. Postel, September 1981, http://www.internet.nic

[8] RFC 822, *Standard For The Format of ARPA Internet Text Messages*, D.H.Crocker, August 1982, , http://www.internet.nic .

Please Note:

Referenced documents and other documents related to TMX Information Processor products can be retrieved from the TMX Document portal at https://www.tcbdata.com/tmxequitymarkets/login.cfm .

7. Data Dictionary

A

AlphaNumeric – alphabetic and numeric characters.

```
      AlphaNumeric = all US-ASCII character, 0x00 to 0x7f
```

B

BusinessAction – the action to take for a BusinessContent section.

```
      FieldIdentifier = 5   ;    no default   Maximum 35 Characters
      BusinessAction   = ''Quote''          |
```

BusinessClass – the message class for a Business Content Layer message.

```
      FieldIdentifier = 6 ; no default     Maximum 35 Characters
      BusinessClass    = ''Quote''         |
```

BusinessContent – the business fields for a STAMP message.

```
      BusinessContent = BusinessContentChar 1*BusinessContentField
```

BusinessContentChar – the character that introduces BusinessContent.

```
      BusinessContentChar = <US-ASCII FS; File Separator>  ;  0x1c
```

BusinessContentField – a field found in the Business Content section of a message.

C

CdfId – Unique internal identifier which includes an internal sequence number assigned by the system to each CDF message for tracking and audit.

```
      FieldIdentifier = 513   ;    no default
      CdfId = 1*31 AlphaNumeric
```

CdfInboundTimeStamp – Unique internal inbound CDF consolidation timestamp assigned by the system to each CDF message for tracking and audit.

```
      FieldIdentifier     = 515   ;  no default
      CdfInboundTimeStamp = 17Digit  ;  YYYYMMDDHHMMSSmmm (year, month, day,
                                        hour, minute, second, millisecond)
```

CdfOutboundTimeStamp – Unique internal outbound CDF consolidation timestamp assigned by the system to each CDF message for tracking and audit.

```
      FieldIdentifier      = 514   ;  no default
      CdfOutboundTimeStamp = 17Digit  ;  YYYYMMDDHHMMSSmmm (year, month, day,
                                         hour, minute, second, millisecond)
```

CdfPubTimeStamp – the time at which the CDF message was sent.

```
      FieldIdentifier = 501  ;  no default
      CdfPubTimeStamp = 17Digit  ;  YYYYMMDDHHMMSSmmm (year, month, day,
                                    hour, minute, second, millisecond)
```

CdfRcvTimeStamp – the time at which the CDF message was received.

```
      FieldIdentifier = 502  ;  no default
      CdfRcvTimeStamp = 17Digit  ;  YYYYMMDDHHMMSSmmm (year, month, day,
                                    hour, minute, second, millisecond)
```

ControlHeader – the portion of the STAMP message that contains administrative information.

```
      ControlHeader = ControlHeaderChar 1*ControlHeaderField
```

ControlHeaderChar – the character that introduces ControlHeader.

```
      ControlHeaderChar = <US-ASCII SOH; Start of Heading>  ;  0x01
```

ControlHeaderField – a field found in the ControlHeader section of a message.

D

Date – the date format.

```
      Date = 8Digit  ;   in YYYYMMDD format
```

DestAddress – the destination STAMP address.

```
      FieldIdentifier = 17
      DestAddress     = DirectedAddress | BroadcastAddress  ;  no default
```

Note that only servers are allowed to use `BroadcastAddress`.

Digit – representation of numeric values.

```
      Digit = ‘‘0’’ | ‘‘1’’ | ‘‘2’’ | ‘‘3’’ | ‘‘4’’ | ‘‘5’’ | ‘‘6’’ | ‘‘7’’ | ‘‘8’’ |
      ‘‘9’’
```

DirectedAddress – a specific STAMP address.

```
      DirectedAddress = 8Hexadecimal  ;   4 bytes (00000000 is reserved)
```

The value is a 4 byte value encoded in 8 byte hexadecimal format. Please refer to the *STAMP Assigned Addresses* document for details as to how these numbers are assigned.

E

Empty – nothing.

```
      Empty = ‘‘’’
```

ExchangeId –identifies the exchange from which the message originated.

```
      FieldIdentifier = 247
       ExchangeId      = ``ALP''- TSX Alpha Exchange          |
                         ``CDX''- TSXVenture                  |
                         ``CHI''- Chi-XCanada                 |
                         ``CHT''- CX2                         |
                         ``CNQ''- CSE Listed Securities (formerly known as CNSX)|
                         ``ICX''- InstinetCanadaCross         |
                         ``LIQ''- LiquidnetCanada             |
                         ``LYX''- Lynx ATS                    |
                         ``OMG''- OmegaATS                    |
                         ``PUR''- CSE Other Listed Securities (formerly known as
                         PURE Trading)                        |
                         ``SGM''- SigmaX ATS                  |
                         ``TCM''- TriActMatchNow              |
                         ``TSE''- TSX                         |
                         ``AQL'' - Aequitas - Lit             |
                         ``AQN ''- Aequitas - Neo             |
```

F

Field – a unit within a section that includes a FieldIdentifier and an optional FieldValue.

```
      Field = FieldChar FieldIdentifier ``='' [FieldValue]
```

FieldChar – the character that introduces a field.

```
      FieldChar = <US-ASCII RS; Record Separator>  ; 0x1e
```

FieldIdentifier – the value that identifies what the field means.

```
      FieldIdentifier = 1*4Digit [FieldIdentifierIndex]  ;    1 to 9999, no default
```

FieldIdentifierIndex – an instance of a specific field within a message.

```
      FieldIdentifierIndex = ``.'' 1*4Digit  ;    0 to 9999, default is 0
```

FieldValue – the value of the field.

```
      FieldValue = 1*PrintableChar
```

H

Hexadecimal – hexadecimal number representation.

```
      Hexadecimal = Digit | ``a'' | ``b'' | ``c'' | ``d'' | ``e'' | ``f''
```

I

L

M

N

NumericPrice – a price in a currency.

```
      NumericPrice = 1*6Digit [‘‘.’’ 1*5Digit]
```

O

P

PrintableASCII – characters that have a glyph from the US-ASCII character set.

```
      PrintableASCII = <any printable char from US-ASCII char set plus HT>
                       ; 0x09, 0x20 to 0x3c, 0x3e to 0x7e
```

PrintableChar – characters that have a glyph.

```
      PrintableChar = PrintableASCII | PrintableLatin1
```

PrintableLatin1 – characters that have a glyph from the Latin 1 character set.

```
      PrintableLatin1 = <any printable char from Latin 1 char set> ; 0xa1 to 0xff
```

PublicPrice – the public price of an order (specifically different than the ‘private’ price for some pre-open orders).

```
       FieldIdentifier = 196

       PublicPrice   = Price ; no default
```

R

Retrans – a marker that indicates the message is a retransmitted message.

```
      FieldIdentifier = 97
      Retrans         = ‘‘Y’’| ‘‘N’’   ;  default is ‘‘N’’
```

RetransId – an identifier as to which retransmission request caused the retransmission.

```
      FieldIdentifier = 147
      RetransId       = 1*5AlphaNumeric  ; no default
```

S

SequenceNumber – the sequence number of the message.

```
      FieldIdentifier = 50
      SequenceNumber  = 1*9Digit  ;  0 to 999,999,999  ; no default
```

SourceAddress – the source STAMP address.

```
      FieldIdentifier = 54
      SourceAddress   = DirectedAddress  ;  no default
```

Symbol – the security/issue symbol.

```
      FieldIdentifier = 55
      Symbol          = 1*17AlphaNumeric   ;   no default
```

T

V

Volume – the quantity of shares for an order or a fill report.

```
      FieldIdentifier = 64
      Volume          = 1*10Digit   ;   no default
```

8. Field Identifiders by Numerical Order

5	BusinessAction
6	BusinessClass
17	DestAddress
50	SequenceNumber
54	SourceAddress
55	Symbol
64	Volume
97	Retrans
147	RetransId
196	PublicPrice
247	ExchangeId
501	CdfPubTimeStamp
502	CdfRcvTimeStamp
513	CdfId
514	CdfOutboundTimeStamp
515	CdfInboundTimeStamp

9. Field Identifiers by Alphabetical Order

5	BusinessAction
6	BusinessClass
513	CdfId
515	CdfInboundTimeStamp
514	CdfOutboundTimeStamp
501	CdfPubTimeStamp
502	CdfRcvTimeStamp
17	DestAddress
247	ExchangeId
196	PublicPrice
97	Retrans
147	RetransId
166	SellParticipation
50	SequenceNumber
53	SettlementTerms
54	SourceAddress
199	SpecialistName
312	SpecialistPhoneNumber
282	StockGroup
80	StockHaltDate
161	StockState
55	Symbol
64	Volume

TMX Group
The Exchange Tower
130 King St. W.
Toronto, ON M5X 1J2

TMX Vendor Services
vendor_services@tmx.com
416-947-4705