QRTMD TSX and TSXV Level 2

TL2/CL2

Functional Specifications

Document Revision: 1.02
Date of Issue: January 8, 2013

Disclaimer and Rights Granted

The information contained in this document and any other information or documentation related to this specification and associated message formats (collectively, along with any upgrades, updates and new versions or releases provided to you by TSX Inc. ("TSX") from time to time, referred to as the "TL2/CL2 Functional Specifications", 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 TL2/CL2 Functional Specifications (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 TL2/CL2 Functional Specifications, you are agreeing o 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 TL2/CL2 Functional Specifications, 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.

Please note that TSX will require prior certification of any system designed to connect to its trading and market information as well as your participation from time to time in connection with the testing of changes or upgrades to the TL2/CL2 Functional Specifications or the systems used to connect to TSX.

The information contained in the TL2/CL2 Functional Specifications 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 in the TL2/CL2 Functional Specifications belong to TSX (or its licensors or service providers, as applicable). Your permitted use of the TL2/CL2 Functional Specifications, 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 trading and market information system. You are not permitted to use all or any part of the TL2/CL2 Functional Specifications for order entry other than on Toronto Stock Exchange, TSX Venture Exchange, TMX Select, or Alpha. 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, commercialize or make any part of the TL2/CL2 Functional Specifications available, in whole or in part, to any person 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 TL2/CL2 Functional Specification in your possession.

Contact Information

For questions, contact TMX Vendor Services, vendor_services@tmx.com, 416-947-4705.

Table of Contents

Chapter 1 Overview

This specification defines the communications interface and the information records transmitted on the Quantum RTMD (QRTMD) Toronto Stock Exchange Level 2 (TL2) and TSX Venture Level 2 (CL2) services.

The TL2 and CL2 services are comprised of:

- Symbol and market status
- Trades
- Orders and confirmations

The Toronto Level 2 and TSX Venture Exchange Level 2 business content messages are formatted using the STAMP protocol syntax. STAMP, the Securities Trading Access Message Protocol, is the messaging protocol developed by the TSX for order entry. More details about the STAMP protocol are given in the *STAMP Specification* (Reference [1]).

To develop and implement systems that can receive and process TL2/CL2 data, this document must be used with the *Quantum RTMD Protocol Specification and Service Access Guide*.

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 Chapter 3, *Business Content Messages,* and Chapter 5, *Field Definitions*. These two sections define how the trading information is defined in the Toronto Level 2 and TSX Venture Exchange Level 2 services. 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 `Consolas typeface` font means that it is a rule that is defined in the *Field Definitions* starting on page 22.
- Any rule that is enclosed in square brackets, "`[`" and "`]`" means that the rule is optional.
- The spaces between the rules mean 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 [7]). 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 *Field Definitions* 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 itself, 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.*

1.2.6 Optional: [Rule]

Square brackets enclose optional elements; for example, "`[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.*

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 Toronto Level 2 or TSX Venture Exchange Level 2 service.

Chapter 2 Message Structure

Business content in TL2 and CL2 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
```

2.1 Control Header Content

```
ControlHeaderContent    =  DestAddress SequenceNumber TimeStamp
                           [LastSequenceReceived] [Retrans] [RetransId]
                           [SourceAddress]
```

The Toronto Level 2 and TSX Venture Exchange Level 2 services include 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 Toronto Level 2 or TSX Venture Exchange Level 2 service is `TimeStamp`.

The `ControlHeaderChar` (0x01), `BusinessContentChar` (0x1c), and `ControlTrailerChar` (0x1d) separators are not explicitly mentioned in Chapter 3, *Business Content Messages*.

2.2 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 *Field Definitions* 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 *Field Definitions)*.

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 *Field Definitions* 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 either the TL2 or CL2 service.

2.2.1 Field ordering

The order of the fields within a section of a STAMP message is 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.

2.2.2 Field Identifier

The *Field Identifier* is a number that is used as an index into the *Field Definitions* 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` (see Section 3.3 on page 15) 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 as follows: 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
```

2.2.3 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
```

2.2.4 PrivateKeyIdentifier tag handling

The BusinessContent of some TL2 and CL2 messages will include the PrivateKeyIdentifi0er field (tag 165). Tag 165 is a leftover after separating the public data part of the feed prior to dissemination. When present, this tag should be ignored. If the message's BusinessContent consists of only the PrivateKeyIdentifier field, it should be ignored.

Chapter 3 Business Content Messages

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

3.1 Trading Tier Status Message

TSX and TSXV send Trading Tier Status messages to the Client at the beginning of each trading day. The Trading Tier Status message provides statistical information to the trading community about the total number of:

- Stock groups per Trading Engine for the trading day.
- Symbols per Trading Engine for the trading day.
- Open orders per Trading Engine for the trading day.

TradingTierStatus Message	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessAction BusinessClass ExchangeId TotalNumOpenOrders TotalNumStockGroups TotalNumSymbols TradingSysTimeStamp TradingTierId
Where:		
BusinessClass	=	"MarketInfo"
BusinessAction	=	"TradingTierStatus"

3.2 SymbolStatus Message

SymbolStatus message provides information for an equity, debenture or trading instrument for the current trading day.

- The SymbolStatus message is available at the beginning of each trading day.
- SymbolStatus message provides information for an equity, debenture or trading instrument for the current trading day.
- The ProductType field indicates an equity or debenture security.
- The LastMessage tag indicates the completion for the Symbol Status messages.

SymbolStatusMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessAction BusinessClass Symbol TradingSysTimeStamp [AcceptAnonymous] [AcceptUndisplayed] [BoardLot] [Currency] [CUSIP] [ExchangeId] [FaceValue] [LastMessage] [LastSale] [MGF-Setting] [MGF-Volume] [MocEligible] [NumberOfMessages] [OpeningTime] [ProductType] [SpecialistName] [SpecialistPhoneNumber] [StockGroup] [StockHaltDate] [StockState] [SymbolFullName] [TotalNumMessages]
Where:		
BusinessClass	=	“SymbolInfo”
BusinessAction	=	“SymbolStatus”

3.3 OrderBook Message

OrderBook message provides the public information for all open orders in the market. The information is as of the end of the previous business day. Its purpose is to enable the initialization of the Client's trading book for the current trading session.

- The `OrderBook` message is available at the beginning of each trading day.
- The `LastMessage` tag indicates the completion for the OrderBook messages.

OrderBookMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessAction BusinessClass TradingSysTimeStamp [BrokerNumber] [ExchangeId] [LastMessage] [MarketSide] [NonResident] [NumberOfMessages] [OrderNumber] [PrivateKeyIdentifier] [PriorityTimeStamp] [PublicPrice] [SettlementTerms] [Symbol] [TotalNumMessages] [Volume]
Where:		
BusinessClass	=	"OrderInfo"
BusinessAction	=	"OrderBook"

3.4 Order/Cancel Confirmation Report

TSX and TSXV send an Order/Cancel Confirmation Report in response to a new order or cancel being entered into the trading system.

An Order/Cancel Confirmation Report will confirm changes to the attributes of an existing order when they have been modified.

OrderCancelRespMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
PublicContent	=	BrokerNumber BusinessAction BusinessClass ConfirmationType OrderNumber PublicPrice Symbol TradingSysTimeStamp Volume [ByPass] [CFOdOrderNumber] [ExchangeId] [NonResident] [PriorityTimeStamp] [PriorityVolume] [SettlementTerms]
BusinessClass	=	“OrderCancelResp”
BusinessAction	=	“Buy” | “Sell”
ConfirmationType	=	AssignTimePriority | Booked | Cancelled | PriceAssigned

The following are the meanings of the `ConfirmationType` field:

“AssignTimePriority”	The order has been sequenced in the book according to its time priority. The `PriorityTimeStamp` indicates the new time stamp used for sequencing the order in the book.
“Booked”	The order has been entered in the book and is now eligible for matching.
“Cancelled”	The order has been cancelled by the submitting broker and is no longer in the book.
“PriceAssigned”	The order has been re-priced.

Note: CL2 does not support the following tags: `NonResident`.

3.5 Trade Report

TSX and TSXV send a Trade Report in response to a trade occurring on a previously accepted new order, CFO, or cross.

The Trade Report includes all relevant transaction details including the opposite broker number, remaining order volume and if the member acted as principal on the trade.

TradeReportMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	2BrokerNumber BusinessAction BusinessClass 2OrderNumber Price Symbol TradeNumber TradingSysTimeStamp Volume [ByPass] [CFOdOrderNumber] [CrossType] [2DisplayVolume] [ExchangeId] [ExtendedHours] [LastSale] [Moc] [NonResident] [OrigTradeID] [2PriorityTimeStamp] [SettlementTerms] [TradeCorrection] [TradeTimeStamp]
Where:		
BusinessClass	=	“TradeReport”
BusinessAction	=	“Cancelled” | “Trade”

Each trade consists of two fills. 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.

BusinessAction of “Cancelled” will appear in the Trade Report when the TSX Trading Services officer cancels a trade after it has been transacted.

Note: CL2 does not support `NonResident`.

3.6 Stock Status Notification

TSX and TSXV sends a Stock Status Notification in response to a change in stock status on the Trading Engine.

StockStatusMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass Symbol TradingSysTimeStamp [AcceptAnonymous] [AcceptUndisplayed] [BlindOffsetAccepted] [CalculatedClosingPrice] [Comment] [Currency] [ExchangeId] [MGF-Volume] [MocEligible] [MocVwap] [OpeningTime] [SpecialistName] [SpecialistPhoneNumber] [StockState]
Where:		
BusinessClass	=	“StockStatus”

3.7 Market State Change

TSX and TSXV send Market State Change message whenever a notice of a market state change or a trading session change has been received from the Trading Engine.

MarketStateChangeMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass TradingSysTimeStamp [ExchangeId] [MarketState] [StockGroup]
Where:		
BusinessClass	=	“MarketStateChange”

3.8 General Message

TSX and TSXV send a General message when it is generated by the Trading Engine.

GeneralMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass MessageText TradingSysTimeStamp [BulletinIndicator] [ExchangeId]
Where:		
BusinessClass	=	“GeneralMessage”

3.9 MBX Message

MBX is a generic name for “Market By Order” or “Market By Price”.

The MBX message provides additional information, beyond orders and fills, to a gateway building MBX displays. A `BusinessAction` of **“**`AssignCOP`**”** is used when specifying the Calculated Opening Price (COP), and the list of Participating `OrderKeys` that are not priced at the COP. A `BusinessAction` of **“**`AssignLimit`**”** is used when resetting former better-priced-limit orders to their true limits.

A complete MBX message may be broken into parts for transmission if the message is very large. This is handled using the `MBX_PartNumber` and `MBX_TotalParts` fields. These fields will only be present if the message has been broken into parts.

MBXMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SourceAddress SequenceNumber TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessAction BusinessClass CalculatedOpeningPrice Symbol TradingSysTimeStamp [ExchangeId] [MBX_PartNumber] [MBX_TotalParts] [1*OrderKey] [1*Price]
Where:		
BusinessClass	=	“MBXMessage”
BusinessAction	=	“AssignCOP” | “AssignLimit”

Note: MBX messages exceeding 1,400 characters in length will be delivered on multiple packets using the header Continuation Indicator described in the *Quantum RTMD Protocol Specification and Service Access Guide*. For an MBX message, broken parts will not generally be in a stand alone packet. All packets corresponding to a single logical MBX message must be reassembled by the receiver before attempting to parse the STAMP tags inside the message.

3.10 MOC Imbalance Notification

Note: The TSX (TL2) or TSXV (CL2) Server sends a MOC Imbalance Notification to the Client for MOC-eligible symbols at 3:40 p.m.

MocImbalanceMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass ExchangeId Symbol TradingSysTimeStamp [ImbalanceSide] [ImbalanceVolume]
Where:		
BusinessClass	=	"MocImbalanceStatus"

Chapter 4 Operating Sequence

4.1 Transmission times

Clients can listen on the TL2 or CL2 port at any time during the day. The unsequenced Heartbeat message is transmitted every 60 seconds.

Start of the day TradingTierStatus, StockStatus and OrderBook messages are sent at the beginning of each trading day.

Transmission times for TL2 and CL2 are Eastern Standard/Daylight Savings Time.

Please contact TMX Vendor Services for Operational hours.

4.2 Trading hours for the TSX and TSX Venture Exchange

Exchange	Regular Trading Session (EST)			Last Sale Trading Session (EST)		
	Pre-Open	Open	Close	Pre-Open	Open	Close
TSX	07:00	09:30	16:00	N/A	16:15	17:00
TSX Venture Exchange	07:00	09:30	16:00	N/A	16:15	17:00

Note: For MOC rules, session times and eligible securities: Please refer to the TMX Website (http://www.tmx.com) under the Trading/Products and Services/Market on Close section.

Chapter 5 Field Definitions

A

AcceptAnonymous

Flag to indicate if a stock symbol is eligible to accept Anonymous orders.

```
FieldIdentifier = 110          ;
AcceptAnonymous = “Y” | “N”  ;  default is “Y”
```

AcceptUndisplayed

Flag to indicate if a stock symbol is eligible for undisplayed orders.

```
FieldIdentifier = 605          ;
AcceptUndisplayed = “Y” | “N”  ;  default is “Y”
```

AlphaNumeric

Alphabetic and numeric characters.

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

B

BlindOffsetAccepted

Value identifying that MOC Blind Offsetting orders have been accepted.

```
FieldIdentifier     = 490           ;  no default
BlindOffsetAccepted = "OffsetAcpt"
```

BoardLot

Board lot volume.

```
FieldIdentifier = 115
BoardLot        = Volume  ;  no default
```

BrokerNumber

An exchange-assigned number identifying a Member Firm.

```
FieldIdentifier = 70
BrokerNumber    = 1*3Digit  ;  no default
```

BulletinIndicator

Indicates message is a bulletin.

```
FieldIdentifier   = 317
BulletinIndicator = “Y” | “N”  ;  default is “N”
```

BusinessAction

The action to take for a BusinessContent section.

```
FieldIdentifier = 5  ;  no default    Maximum 35 Characters
BusinessAction  = "AssignCOP"         |
                  "AssignLimit"       |
                  "Buy"               |
                  "Cancelled"         |
                  "Cross"             |
                  "DelayOpenStock"    |
                  "OpenDelayedStock"  |
                  "Sell"              |
                  "Trade"             |
                  "SymbolStatus"      |
                  "OrderBook"         |
                  "TradingTierStatus" |
```

BusinessClass

The message class for a Business Content Layer message.

```
FieldIdentifier = 6  ;  no default      Maximum 35 Characters
BusinessClass   = "GeneralMessage"      |
                  "MarketStateChange"   |
                  "MBXMessage"          |
                  "OrderCancelResp"     |
                  "StockStatus"         |
                  "TradeReport"         |
                  "MocImbalanceStatus"  |
                  "MarketInfo"          |
                  "SymbolInfo"          |
                  "OrderInfo"           |
```

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.

ByPass

To indicate orders are tradable against only visible/disclosed volumes and bypasses iceberg orders, RT participation and autofill, and special terms book. Any part of the OrderQty balance not filled immediately is "killed/cancelled".

```
FieldIdentifier = 503  ;
ByPass          = "Y" | "N"  ;  default "N"
```

C

CalculatedClosingPrice

The price at which MOC orders will trade at Closing.

```
FieldIdentifier        = 491
CalculatedClosingPrice = Price
```

CalculatedOpeningPrice

The price at which orders will trade at the opening.

```
FieldIdentifier        = 191
CalculatedOpeningPrice = Price  ;  no default
```

CFOdOrderNumber

The original order number of the order that was CFOd.

```
FieldIdentifier = 11
CFOdOrderNumber = OrderNumber  ;  no default;
```

Comment

A text field corresponding to a reason code entered by Market Surveillance when a stock is halted; or, the initiator of a delayed opening on a stock, or when there is a change to the RT/Oddlot Trader on a stock. As well, this is a system generated text field to describe the disabling of the MOC Session by TSX Trading Services.

```
FieldIdentifier = 173
Comment         = 1*40AlphaNumeric  ;  no default
```

Note:

1. When a Primary RT is added to a stock (from no RT to having an RT) – Comment = “RT Change”
2. When a Primary RT exists for a stock and is changed – Comment = “RT Change”
3. When a Primary RT is removed from a stock (leaving no RT) – Comment = “RT Removed”

Notes for MOC-Eligible stock:

MOC Disabled, See Trader Notes for Details

ConfirmationType

The type of confirmation for a report.

```
FieldIdentifier  = 16
ConfirmationType = “AssignTimePriority”  |
                   “Booked”              |
                   “Cancelled”           |
                   “PriceAssigned”       |
```

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.

CrossType

Type of crosses originating from a Participating Organization between managed accounts that have the same manager.

```
FieldIdentifier = 390  ;  no default
CrossType = “Basis”  ; Basis (TSX Only)                          |
            “Contgt” ; Contingent (TSX Only)                     |
            “Intrnl” ; Internal                                  |
            “STS”    ; Special Trading Session (TSX Only)       |
            “VWAP”   ; Volume Weighted Average Price (TSX Only)
```

Currency

The currency of a price.

```
FieldIdentifier = 58
Currency        = "CAD"  ; default  |
                  "USD"
```

CUSIP

Clearing and settlement registration number.

```
FieldIdentifier = 171
CUSIP           = 9*12AlphaNumeric  ;  no default
```

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.

DisplayVolume

Public remaining volume.

```
FieldIdentifier = 150
DisplayVolume   = Volume
```

E

Empty

Nothing.

```
Empty = “”
```

ExchangeId

Identifies the exchange from which the message originated.

```
FieldIdentifier = 247
ExchangeId      = “CDX” |
                  “CD1” |
                  “CD2” |
                  “CD3” |
                  “TSE” |
                  “TS1” |
                  “TS2” |
                  “TS3” |
```

ExtendedHours

To indicate action occurred during extended hours session (the Last Sale Trading Session).

```
FieldIdentifier = 76
ExtendedHours   = “Y” | “N”  ;  default is “N”
```

F

FaceValue

The face value of a debenture.

```
FieldIdentifier = 119
FaceValue       = Price  ;  no default
```

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

ImbalanceSide

Marker to indicate which side has a Volume Imbalance for Market On Close.

```
FieldIdentifier = 492  ;  no default
ImbalanceSide   = “BuySide”     |
                  “SellSide”    |
                  “NA”  ;  used when no Imbalance exists
```

ImbalanceVolume

Identifies the volume of shares of the Imbalance side for Market On Close.

```
      FieldIdentifier = 493
ImbalanceVolume = 1*9 digit  ;  no default
```

L

LastMessage

A marker to indicate that the current query response is the last in a series. It varies, depending on the type of message:

SymbolStatus message: Set the `LastMessage` indicator to ‘Y’ on the last symbol message in the stock group

OrderBook message: Set the `LastMessage` indicator to ‘Y’ on the last open order message in the stock group

```
FieldIdentifier = 113
LastMessage     = “Y” | “N”  ;  default is “N”
```

LastSale

Last sale price of a stock.

```
FieldIdentifier = 114
LastSale        = NumericPrice  ;  no default
```

LastSequenceReceived

The last sequence number received.

```
FieldIdentifier    = 15
LastSequenceNumber = SequenceNumber  ;  no default
```

M

MarketSide

The buy or sell side of the market.

```
FieldIdentifier = 197
MarketSide      = “Buy” | “Sell”  ;  no default
```

MarketState

The indication of the current market state.

```
FieldIdentifier = 159
MarketState     = "Pre-open"                |
                  "Opening"                 |
                  "Open"                    |
                  "Closed"                  |
                  "Extended Hours Open"     |
                  "Extended Hours Close"    |
                  "Extended Hours CXLs"     |
                  "MOC Imbalance"           |
                  "CCP Determination"       |
                  "PriceMovementExtension"  |
                  "Closing"                 |
```

MBX_PartNumber

Number identifying an MBX part message.

```
FieldIdentifier = 194
MBX_PartNumber  = 1*3Digit  ;  no default
```

MBX_TotalParts

Total number of parts in a fragmented MBX message.

```
FieldIdentifier = 195
MBX_TotalParts  = 1*3Digit  ;  no default
```

MessageText

The description for a trading system generated message.

```
FieldIdentifier = 160
MessageText     = 1*1024PrintableChar  ;  no default
```

MGF-Setting

Indicator to show if Minimum Guaranteed Fill processing is activated for this Symbol.

```
FieldIdentifier = 284          ;  TSX only
MGF-Setting     = "On" | "Off" ;  no default
```

MGF-Volume

The Minimum Guaranteed Fill volume.

```
FieldIdentifier = 49
MGF-Volume      = Volume  ;  no default
```

Moc

Identifies the trade as a Market on Close trade.

```
FieldIdentifier = 494
Moc             = "Y" | "N"  ;  default is "N"
```

MocEligible

Identifies whether the stock is eligible to participate in the Market on Close session each day.

```
Field Identifier = 496
MocEligible      = "Y" | "N"
```

MocVwap

Volume weighted average price based on trades occurring in the continuous market for MOC.

```
FieldIdentifier = 495
MocVwap       = Price
```

N

NonResident

A terms marker indicating that trade participant is not a Canadian resident.

```
FieldIdentifier = 168
NonResident     = "Y" | "N"  ;  default is "N"
```

NumberOfMessages

Number of messages contained in the query response.

Symbol Status message: Assign incremental numbers for each symbol message in the stock group

Order Book message: Assign incremental numbers for each open order message in the stock group

```
FieldIdentifier  = 111
NumberOfMessages = 1*8Digit  ;  no default
```

NumericPrice

A price in a currency.

```
NumericPrice = 1*6Digit ["." 1*5Digit]
```

O

Opening Time

Contained in the Symbol Status.

```
Field Identifier = 120
```

OrderKey

Unique key identifying orders in the system.

```
FieldIdentifier = 192
OrderKey        = BrokerNumber "|" OrderNumber  ;  no default
```

OrderNumber

A number assigned to the order by the trading system.

```
FieldIdentifier = 40                ;  no default
OrderNumber     = 1*18AlphaNumeric
```

OrigTradeID

Used with trade corrections to reference previously reported executions and the side initiating the cancel/correct

```
FieldIdentifier = 506 ;
OrigTradeID = Trade # | Side  (B=Buy; S=Sell  ;  C = combined Indicator for
                                                 both sides) |
```

P

Price

The limit or type of price for an order.

```
FieldIdentifier = 41
Price           = NumericPrice                      |
                  "MKT"  ;  Market Price, default   |
                  "OPG"  ;  Opening Price           |
                  "MBF"  ;  Must Be Filled
```

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
```

PriorityTimeStamp

Timestamp assigned by the trading engine to specify time priority of an order. Orders are sequenced in the order book based on symbol, price and PriorityTimeStamp.

```
FieldIdentifier   = 178
PriorityTimeStamp = 20Digit ; YYYYMMDDHHMMSSmmmmmm (year, month, day, hour,
                              minute, second, millionths of a second)
```

PriorityVolume

The volume of the order that has priority.

```
FieldIdentifier = 68
PriorityVolume  = Volume  ;  no default
```

PrivateKeyIdentifier

The business fields for a message.

```
FieldIdentifier      = 165
PrivateKeyIdentifier = 1*4Digit
```

The `PrivateKeyIdentifier` identifies the entity that has the necessary key codes to decrypt the corresponding `PrivateContent` in a message (if present this tag should be ignored as it is not applicable to TL2 messages.)

ProductType

The product type for a symbol.

```
FieldIdentifier = 105
ProductType     = "CDNX-Equity"     |  ;
                  "CDNX-Debenture   |  ;
                  "Debenture"       |
                  "Equity"
```

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
```

SettlementTerms

The terms for settlement of the order.

```
FieldIdentifier = 53      ; no default
SettlementTerms = "Cash"  |
                  "CT"    ; cash today  |
                  Date    ; delayed delivery date  |
                 "MS"     ; derivatives-related contingent equity trade |
                 "NN"     ; non-net
```

SourceAddress

The source STAMP address.

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

SpecialistName

The stock specialist's full name.

```
FieldIdentifier = 199
SpecialistName  = 1*30AlphaNumeric  ;  default is none
```

SpecialistPhoneNumber

Phone number for the Specialist trader for the stock.

```
FieldIdentifier       = 312
SpecialistPhoneNumber = 1*30AlphaNumeric  ;  default is none
```

StockGroup

An indicator of stock group.

```
FieldIdentifier = 282
StockGroup      = 1*2Digit  ;  no default
```

StockHaltDate

The date on which the stock was halted.

```
FieldIdentifier = 80
StockHaltDate   = Date  ;  no default
```

StockState

The possible states that a stock may be in that are broadcast.

```
FieldIdentifier = 161
StockState     =  "Authorized"                     |
                  "AuthorizedDelayed"              |
                  "AuthorizedFrozen"               |
                  "AuthorizedHalted"               |
                  "Inhibited"                      |
                  "InhibitedDelayed"               |
                  "InhibitedFrozen"                |
                  "InhibitedHalted"                |
                  "AuthorizedPriceMovementDelayed" |
                  "AuthorizedPriceMovementFrozen"  |
                  "InhibitedPriceMovementDelay"    |
                  "InhibitedPriceMovementFrozen"
```

Symbol

The security/issue symbol.

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

SymbolFullName

The security/issue symbol's complete company name.

```
FieldIdentifier = 177
SymbolFullName  = 1*80PrintableASCII  ;  no default
```

T

TimeStamp

The time at which the STAMP message was sent.

```
FieldIdentifier = 56         ; no default
TimeStamp       = 16Digit    ; YYYYMMDDHHMMSShh (year, month, day, hour
                             ; minute, second, hundredths of a second)
```

Note that for a retransmitted message, the value of TimeStamp is the time of the retransmission, not the transmission time of the original message.

TotalNumMessages

The total number of messages expected for a query response.

- **Symbol Status** message: Total number of symbols per stock group
- **Order Book** message: Total number of open orders per stock group

```
FieldIdentifier  = 112
TotalNumMessages = 1*8Digit  ;  no default
```

TotalNumOpenOrders

The total number of open orders per trading tier and ExchangeId.

```
FieldIdentifier    = 581       ;
TotalNumOpenOrders = 1*7Digit  ;  0 to 9,999,999
```

TotalNumStockGroups

The total number of stock groups per trading tier and ExchangeId.

```
FieldIdentifier     = 582  ;
TotalNumStockGroups = 1*3Digit
```

TotalNumSymbols

The total number of stocks per trading tier and ExchangeId.

```
FieldIdentifier = 583  ;
TotalNumSymbols = 1*5Digit
```

TradeCorrection

An indicator as to whether the Trade Report is a trade correction or a normal fill.

```
FieldIdentifier = 183
TradeCorrection = “Y” | “N”  ;  default is “N”
```

TradeNumber

Unique identifier assigned to each trade on a per stock basis.

```
FieldIdentifier = 220
TradeNumber     = 1*7Digit  ;  no default
```

TradeTimeStamp

The time at which the trade occurred, manually set when a trade is added by TSX or TSXV.

```
FieldIdentifier = 264
TradeTimeStamp  = 16Digit    ; YYYYMMDDHHMMSShh (year, month, day, hour
                             ; minute, second, hundredths of a second)
```

TradingSysTimeStamp

The time at which the BusinessAction occurred.

```
FieldIdentifier     = 57
TradingSysTimeStamp = 16Digit    ; YYYYMMDDHHMMSShh (year, month, day, hour
                                 ; minute, second, hundredths of a second)
```

TradingTierId

A Trading Engine identifier.

```
FieldIdentifier = 584  ;
TradingTierId   = 1*9Alphanumeric
```

V

Volume

The quantity of shares for an order or a fill report.

```
FieldIdentifier = 64
Volume          = 1*9Digit  ;  no default
```

Chapter 6 Field Identifiers by Numerical Order

1	RESERVED	40	OrderNumber
2	RESERVED	41	Price
3	RESERVED	42	RESERVED
4	RESERVED	43	RESERVED
5	BusinessAction	44	RESERVED
6	BusinessClass	45	RESERVED
7	RESERVED	46	RESERVED
8	RESERVED	47	RESERVED
9	RESERVED	48	RESERVED
10	RESERVED	49	MGF-Volume
11	CFOdOrderNumber	50	SequenceNumber
12	RESERVED	51	RESERVED
13	RESERVED	52	RESERVED
14	RESERVED	53	SettlementTerms
15	LastSequenceReceived	54	SourceAddress
16	ConfirmationType	55	Symbol
17	DestAddress	56	TimeStamp
18	RESERVED	57	TradingSysTimeStamp
19	RESERVED	58	Currency
20	RESERVED	59	RESERVED
21	RESERVED	60	RESERVED
22	RESERVED	61	RESERVED
23	RESERVED	62	RESERVED
24	RESERVED	63	RESERVED
25	RESERVED	64	Volume
26	RESERVED	65	RESERVED
27	RESERVED	67	RESERVED
28	RESERVED	68	PriorityVolume
29	RESERVED	69	RESERVED
30	RESERVED	70	BrokerNumber
31	MinimumFillVolume	71	RESERVED
32	RESERVED	72	RESERVED
33	RESERVED	73	RESERVED
34	RESERVED	74	LotsOf
35	RESERVED	75	RESERVED
36	RESERVED	76	ExtendedHours
37	RESERVED	77	RESERVED
38	RESERVED	78	RESERVED
39	RESERVED	79	RESERVED

80	StockHaltDate
81	RESERVED
82	RESERVED
83	RESERVED
84	RESERVED
85	RESERVED
86	RESERVED
87	RESERVED
88	RESERVED
89	RESERVED
90	RESERVED
91	RESERVED
92	RESERVED
93	RESERVED
94	RESERVED
95	RESERVED
96	RESERVED
97	Retrans
98	RESERVED
99	RESERVED
100	RESERVED
101	RESERVED
102	RESERVED
103	RESERVED
104	RESERVED
105	ProductType
106	RESERVED
107	RESERVED
108	RESERVED
109	RESERVED
110	AcceptAnonymous
111	NumberOfMessages
112	TotalNumMessages
113	LastMessage
114	LastSale
115	BoardLot
116	RESERVED
117	EquityStatus
118	RESERVED
119	FaceValue
120	OpeningTime
121	RESERVED
122	RESERVED
123	RESERVED
124	RESERVED
125	RESERVED
126	RESERVED
127	RESERVED
128	RESERVED
129	RESERVED
130	RESERVED
131	RESERVED
132	RESERVED
133	RESERVED
134	RESERVED
135	RESERVED
136	RESERVED
137	RESERVED
138	RESERVED
139	RESERVED
140	RESERVED
141	RESERVED
142	RESERVED
143	RESERVED
144	RESERVED
145	RESERVED
146	RESERVED
147	RetransId
148	RESERVED
149	RESERVED
150	Display Volume
151	RESERVED
152	RESERVED
153	RESERVED
154	RESERVED
155	RESERVED
156	RESERVED
157	RESERVED
158	RESERVED
159	MarketState
160	MessageText
161	StockState
162	RESERVED
163	RESERVED

164	RESERVED
165	PrivateKeyIdentifier
166	DELETED
167	DELETED
168	NonResident
169	RESERVED
170	RESERVED
171	CUSIP
172	RESERVED
173	Comment
174	RESERVED
177	SymbolFullName
178	PriorityTimeStamp
179	RESERVED
180	RESERVED
181	RESERVED
182	RESERVED
183	TradeCorrection
184	RESERVED
185	RESERVED
186	RESERVED
187	RESERVED
188	RESERVED
189	RESERVED
190	RESERVED
191	CalculatedOpeningPrice
192	OrderKey
193	RESERVED
194	MBX-PartNumber
195	MBX-TotalParts
196	PublicPrice
197	MarketSide
198	RESERVED
199	SpecialistName
200	RESERVED
201	RESERVED
202	RESERVED
203	RESERVED
204	RESERVED
205	RESERVED
206	RESERVED
207	RESERVED
208	RESERVED
209	RESERVED
210	RESERVED
211	RESERVED
212	RESERVED
213	RESERVED
214	RESERVED
215	RESERVED
216	RESERVED
217	RESERVED
218	RESERVED
219	RESERVED
220	TradeNumber
221	RESERVED
222	RESERVED
223	RESERVED
224	RESERVED
225	RESERVED
226	RESERVED
227	RESERVED
228	RESERVED
229	RESERVED
230	RESERVED
231	RESERVED
232	RESERVED
233	RESERVED
234	RESERVED
235	RESERVED
236	RESERVED
237	RESERVED
238	RESERVED
239	RESERVED
240	RESERVED
241	RESERVED
242	RESERVED
243	RESERVED
244	RESERVED
245	RESERVED
246	RESERVED
247	ExchangeId
248	RESERVED
249	RESERVED

250	RESERVED
251	RESERVED
252	RESERVED
253	RESERVED
254	RESERVED
255	RESERVED
256	RESERVED
257	RESERVED
258	RESERVED
259	RESERVED
260	RESERVED
261	RESERVED
262	RESERVED
263	RESERVED
264	TradeTimeStamp
265	RESERVED
266	RESERVED
267	RESERVED
268	RESERVED
269	RESERVED
270	RESERVED
271	RESERVED
272	RESERVED
273	RESERVED
274	RESERVED
275	RESERVED
276	RESERVED
277	RESERVED
278	RESERVED
279	RESERVED
280	RESERVED
281	RESERVED
282	StockGroup
283	RESERVED
284	MGF-Setting
285	RESERVED
286	RESERVED
287	RESERVED
288	RESERVED
289	RESERVED
290	RESERVED
291	RESERVED
292	RESERVED
293	RESERVED
294	RESERVED
295	RESERVED
296	RESERVED
297	RESERVED
298	RESERVED
299	RESERVED
300	RESERVED
301	RESERVED
302	RESERVED
303	RESERVED
304	RESERVED
305	RESERVED
306	RESERVED
307	RESERVED
308	RESERVED
309	RESERVED
310	RESERVED
311	RESERVED
312	SpecialistPhoneNumber
313	RESERVED
314	RESERVED
315	RESERVED
316	RESERVED
317	BulletinIndicator
318	RESERVED
319	RESERVED
320	RESERVED
321	RESERVED
322	RESERVED
323	RESERVED
324	RESERVED
325	RESERVED
326	RESERVED
327	RESERVED
328	RESERVED
329	RESERVED
330	RESERVED
331	RESERVED
332	RESERVED
333	RESERVED

334	RESERVED
335	RESERVED
336	RESERVED
337	RESERVED
338	RESERVED
339	RESERVED
340	RESERVED
341	RESERVED
342	RESERVED
343	RESERVED
344	RESERVED
345	RESERVED
346	RESERVED
347	RESERVED
348	RESERVED
349	RESERVED
350	RESERVED
351	RESERVED
352	RESERVED
353	RESERVED
354	RESERVED
355	RESERVED
356	RESERVED
357	RESERVED
358	RESERVED
359	RESERVED
360	RESERVED
361	RESERVED
362	RESERVED
363	RESERVED
364	RESERVED
365	RESERVED
366	RESERVED
367	RESERVED
368	RESERVED
369	RESERVED
370	RESERVED
371	RESERVED
372	RESERVED
373	RESERVED
374	RESERVED
375	RESERVED
376	RESERVED
377	RESERVED
378	RESERVED
379	RESERVED
380	RESERVED
381	RESERVED
382	RESERVED
383	RESERVED
384	RESERVED
385	RESERVED
386	RESERVED
387	RESERVED
388	RESERVED
389	RESERVED
390	CrossType
391	RESERVED
392	TradeThroughExempt
393	RESERVED
394	RESERVED
395	RESERVED
396	RESERVED
397	RESERVED
398	RESERVED
399	RESERVED
400	RESERVED
401	RESERVED
402	RESERVED
403	RESERVED
404	RESERVED
405	RESERVED
406	RESERVED
407	RESERVED
408	RESERVED
409	RESERVED
410	RESERVED
411	RESERVED
412	RESERVED
413	RESERVED
414	RESERVED
415	RESERVED
416	RESERVED
417	RESERVED

418	RESERVED
419	RESERVED
420	RESERVED
421	RESERVED
422	RESERVED
423	RESERVED
424	RESERVED
425	RESERVED
426	RESERVED
427	RESERVED
428	RESERVED
429	RESERVED
430	RESERVED
431	RESERVED
432	RESERVED
433	RESERVED
434	RESERVED
435	RESERVED
436	RESERVED
437	RESERVED
438	RESERVED
439	RESERVED
440	RESERVED
441	RESERVED
442	RESERVED
443	RESERVED
444	RESERVED
445	RESERVED
446	RESERVED
447	RESERVED
448	RESERVED
449	RESERVED
450	RESERVED
451	RESERVED
452	RESERVED
453	RESERVED
454	RESERVED
455	RESERVED
456	RESERVED
457	RESERVED
458	RESERVED
459	RESERVED
460	RESERVED
461	RESERVED
462	RESERVED
463	RESERVED
464	RESERVED
465	RESERVED
466	RESERVED
467	RESERVED
468	RESERVED
469	RESERVED
470	RESERVED
471	RESERVED
472	RESERVED
473	RESERVED
474	RESERVED
475	RESERVED
476	RESERVED
477	RESERVED
478	RESERVED
479	RESERVED
480	RESERVED
481	RESERVED
482	RESERVED
483	RESERVED
484	RESERVED
485	RESERVED
486	RESERVED
487	RESERVED
488	RESERVED
489	RESERVED
490	BlindOffsetAccepted
491	CalculatedClosingPrice
492	ImbalanceSide
493	ImbalanceVolume
494	Moc
495	MocVwap
496	MocEligible
497	RESERVED
498	RESERVED
499	RESERVED
500	RESERVED
503	ByPass

506	OrigTradeID
580	DELETED
581	TotalNum`OpenOrders
582	TotalNumStockGroups
583	TotalNumSymbols
584	TradingTierId
585	RESERVED
586	RESERVED
587	RESERVED
588	RESERVED
589	RESERVED
590	RESERVED
591	RESERVED
592	RESERVED
593	RESERVED
594	RESERVED
605	AcceptUndisplayed
606	RESERVED
607	RESERVED
608	RESERVED
609	RESERVED
610	RESERVED
611	RESERVED
612	RESERVED
613	RESERVED
614	RESERVED
615	RESERVED
616	RESERVED
617	RESERVED
618	RESERVED
619	RESERVED
620	RESERVED
621	RESERVED
622	RESERVED

Chapter 7 Field Identifiers by Alphabetical Order

This section lists the subset of STAMP tags (public fields) used in the TL2/CL2 business content.

110	AcceptAnonymous	494	Moc
605	AcceptUndisplayed	496	MocEligible
490	BlindOffsetAccepted	495	MocVwap
115	BoardLot	168	NonResident
70	BrokerNumber	111	NumberOfMessages
317	BulletinIndicator	120	OpeningTime
5	BusinessAction	192	OrderKey
6	BusinessClass	40	OrderNumber
503	ByPass	506	OrigTradeID
191	CalculatedOpeningPrice	41	Price
491	CalculatedClosingPrice	178	PriorityTimeStamp
11	CFOdOrderNumber	68	PriorityVolume
173	Comment	165	PrivateKeyIdentifier
16	ConfirmationType	105	ProductType
390	CrossType	196	PublicPrice
58	Currency	97	Retrans
171	CUSIP	147	RetransId
17	DestAddress	50	SequenceNumber
150	Display Volume	53	SettlementTerms
117	EquityStatus	54	SourceAddress
247	ExchangeId	199	SpecialistName
76	ExtendedHours	312	SpecialistPhoneNumber
119	FaceValue	282	StockGroup
492	ImbalanceSide	80	StockHaltDate
493	ImbalanceVolume	176	StockIndex
113	LastMessage	161	StockState
114	LastSale	55	Symbol
15	LastSequenceReceived	177	SymbolFullName
74	LotsOf	56	TimeStamp
197	MarketSide	112	TotalNumMessages
159	MarketState	581	TotalNumOpenOrders
194	MBX-PartNumber	582	TotalNumStockGroups
195	MBX-TotalParts	583	TotalNumSymbols
160	MessageText	183	TradeCorrection
284	MGF-Setting	220	TradeNumber
49	MGF-Volume	392	TradeThroughExempt
31	MinimumFillVolume	264	TradeTimeStamp

57	TradingSysTimeStamp
584	TradingTierId
64	Volume

Appendix A Symbology Rules

Description	Symbology
Debenture	.DB
Installment	.IR
Preferred Share	.PR
Preferred Share	.PE
Preferred Share	.PF
Rights	.RT
Notes	.NO
Notes	.NS
Notes	.NT
Units	.UN
Warrants	.WT
Class	See table below for Share Classes

A.1 Share Classes

Description	Symbology
Class A, B, C, X	.A, .B, .C
Equity Dividend	.E
NEX (junior venture market) issue	.H
NEX (junior venture market) issue	.K
NEX (junior venture market) issue	.X
Capital Pool Company	.P
Legended Share	.L
BOOMS	.M
Subscription Receipt	.R
Second Subscription Receipts	.N
Third Subscription Receipts	.O
U.S. Trading Terms	.S
Second U.S. Trading Terms	.T
USD Traded Issue	.U
Second USD Traded Issue	.V
When Issued	.W
Second When Issued	.I
Redeemable Common Shares	.Y

A.2 Symbology Notes

Description	Note	Example
All	Symbol Roots can be 1, 2 or 3 digits in length	A – valid AB – valid ABC – valid ABCD – not valid
All	In limited cases, common and preferred shares can have different symbol roots. These cases arise when a company name change is only applicable to common share issues but preferred shares remain under the old company name	TransAlta Corp Common share – TA Preferred share – TAU.PR.A
All	A symbol cannot contain two single letter extensions	ABC.XX.Y – valid ABC.X.Y – not valid
Preferred Share	.PR, .PE, .PF extension must be followed by a second, single letter, extension	ABC.PR.A
Unit	.UN extension cannot be followed by a second, single letter, extension. A second Unit issue for the same issuer will be denoted with a single letter extension	1st Unit Issue – ABC.UN 2nd Unit Issue – ABC.A
Warrant	.WT extension can be followed by a single letter extension for additional warrant issues	1st Warrant – ABC.WT 2nd warrant – ABC.WT.A
Right	.RT extension can be followed by a single letter extension for additional rights issues	1st Right – ABC.RT 2nd Right – ABC.RT.A
Debenture	.DB extension can be followed by a single letter extension for additional debenture issues	1st debenture – ABC.DB 2nd debenture – ABC.DB.A

Appendix B ASCII Character Set

HEX	ASCII	HEX	ASCII	HEX	ASCII
00	NUL	2D	-	5A	Z
01	SOH	2E	.	5B	
02	STX	2F	/	5C	
03	ETX	30	0	5D	
04	EOT	31	1	5E	
05	ENQ	32	2	5F	
06	ACK	33	3		
07	BEL	34	4	5B	
08	BS	35	5	5C	
09	HT	36	6	5D	
0A	LF	37	7	5E	
0B	VT	38	8	60	
0C	FF	39	9	61	a
0D	CR	3A		62	b
0E	SO	3B		63	c
0F	SI	3C		64	d
10	DLE	3D		65	e
11	DC1	3E		66	f
12	DC2	3F		67	g
13	DC3	40	@	68	h
14	DC4	41	A	69	i
15	NAK	42	B	6A	j
16	SYN	43	C	6B	k
17	ETB	44	D	6C	l
18	CAN	45	E	6D	m
19	EM	46	F	6E	n
1A	SUB	47	G	6F	o
1B	ESC	48	H	70	p
1C	FS	49	I	71	q
1D	GS	4A	J	72	r
1E	RS	4B	K	73	s

HEX	ASCII	HEX	ASCII	HEX	ASCII
1F	US	4C	L	74	t
20	SP	4D	M	75	u
21	!	4E	N	76	v
22	"	4F	O	77	w
23	#	50	P	78	y
24	$	51	Q	79	z
25	%	52	R	7A	
26	&	53	S	7B	
27	'	54	T	7C	
28	(	55	U	7D	
29	)	56	V	7E	
2A	*	57	W	7F	DEL
2B	+	58	X		
2C	,	59	Y		

Appendix C Glossary

ARPA	Advanced Research Projects Agency
ASCII	American Standard Code for Information Interchange
BNF	Backus Naur Form
CAD	Canadian dollar
CCP	Calculated Closing Price
CDNX	TSX Venture Exchange
CFO	Change Former Order
CL1	TSX Venture Level 1
CL2	TSX Venture Level 2
COP	Calculated Opening Price
CUSIP	Committee on Uniform Security Identification Procedures
CXL	Cancel
DES	Data Encryption Standard
EST	Eastern Standard Time
ETX	End Of Text
MBF	Must Be Filled
MBX	Market By (Order or Price)
MGF	Minimum Guaranteed Fill
MKT	Market Price
MOC	Market on Close
OPG	Opening Price
RFC	Request for Comment
RT	Registered Trader
QRTMD	Quantum RTMD
STAMP	Securities Trading Access Message Protocol
STX	Start Of Text
TL1	Toronto Level 1
TL2	Toronto Level 2
TMXS	TMX Select
TSX	Toronto Stock Exchange
TSXV	TSX Venture Exchange
USD	United States dollar

VWAP	Volume Weighted Average Price

Appendix D References

[1] *STAMP Specification*, Version 6.4, TMX Group, November 2012.

[2] QRTMD Protocol Specification and Service Access Guide, Version 2.0, TMX Group, December 2012.

[3] QRTM TL1/CL1 Functional Specifications, Version 1.30, TMX Group, January 2013.

[4] *RFC 791, Internet Protocol*, J. Postel, September 1981, http://www.ietf.org.

[5] *RFC 792, Internet Control Message Protocol*, J. Postel, September 1981, http://www.ietf.org.

[6] *RFC 793, Transmission Control Protocol*, J. Postel, September 1981, http://www.ietf.org.

[7] *RFC 822, Standard for the Format of ARPA Internet Text Messages*, David M. Crocker, August 1982, http://www.ietf.org.

[8] *RFC 2117, Protocol Independent Multicast Sparse Mode,* D. Estrin et al, June 1997, http://www.ietf.org.

[9] *RFC 2770, GLOP Addressing in 233/8*, http://www.ietf.org.

Appendix E Revision History

Version	Date	Changes
1.02	2013/01/08	Added information about QRTMD Protocol Specification and Service Access Guide to Introduction. Removed $ from Currency values “CAD” and “USD”.
1.01	2012/06/25	Changed value of TimeStamp and TradingSysTimeStamp to be the same as TradeTimeStamp. Added AuthorizedPriceMovementFrozen value to StockState tag.
1.0	2011/11/01	Initial release.

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

phone: 416.947.4670
toll free: 888.878.8392
fax: 416.947.4662
e-mail: info@tsx.com