Securities Trading and Access Message Protocol (STAMP)

Broadcast Feed Specification

Document Version: 6.42
Publication Date: January 10, 2014

Disclaimer and Rights Granted

The information contained in this document and any other information or documentation related to the STAMP specification and associated message formats (collectively, along with any upgrades, updates and new versions or releases provided to you by TSX Inc. ("TSX") or any of its affiliates from time to time, referred to as the "STAMP 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 STAMP 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 STAMP Specifications, 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 STAMP 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 or any of its affiliates 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 system as well as your participation from time to time in connection with the testing of changes or upgrades to STAMP Specifications or the systems used to connect to TSX.

The information contained in the STAMP 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 STAMP Specifications belong to TSX (or its licensors or service providers, as applicable). Your permitted use of the STAMP 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 STAMP Specifications for order entry other than on Toronto Stock Exchange, TSX Venture Exchange, TMX Select, Montreal Exchange, 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 STAMP 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 STAMP 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

The *Broadcast Feed Specification* describes the messages and architecture of the TSX feeds which broadcast market data. The TSX Broadcast Feed (TBF), TSXV Broadcast Feed (CBF), TMX Select Broadcast Feed (SBF), and Alpha Broadcast Feed (ABF) are based on STAMP-like message architecture. STAMP, the Securities Trading Access Message Protocol, is the messaging protocol developed to support electronic trading.

Note: Unless otherwise noted, the term Broadcast Feed refers to the TBF, CBF, SBF and ABF.

The intention and likelihood is that development of applications using the Broadcast Feeds will be done in conjunction with or as follow-on development to STAMP applications. The Broadcast Feeds are one-way broadcasts from an exchange server to a gateway client. Retransmission requests are handled using the STAMP recovery session. Users of this document should be familiar with the details of message and session architecture described in the *STAMP Specification.*

1.1 Intended audience

This specification is written with four audiences in mind: business analysts, system architects, programmer analysts, and operational 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 5, “Business Content Messages” on page 17 and Chapter 7, “Field Definitions” on page 37. These two sections define how the trading information is defined in the *Broadcast Feed Specification*. In addition to these sections, the business analysts should be familiar with the trading rules and trading scenarios that these messages represent.

System architects will need to examine all aspects of the specification to determine where in their site’s trading system Broadcast Feed functionality is to be implemented. For a more detailed discussion on the layering architecture of the protocol, refer to the *STAMP Specification.*

Programmer analysts should be familiar with the entire specification, although their focus should be on Chapter 2, “Message Architecture” on page 9 and Chapter 3, “Session Architecture” on page 10. In addition to this specification, programmer analysts will require three STAMP Application Notes: *Application Note – STAMP Character Sets, Application Note 2 – The BASE64 Algorithm, and Application Note 3 – The MD5 Algorithm.*

Operational analysts should be familiar with Chapter 3, “Session Architecture” on page 10, paying close attention to session establishment/dissolution and security.

1.2 Rule notation conventions

This section describes the notation convention[1] for the elements of STAMP messages. 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 is the following points:

- Text presented in a `typewriter` typeface means that it is a rule that is defined in the section, "Field Definitions" starting on page 37.
- Optional rules are enclosed in square brackets, `[and]`
- 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*. 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 in-line with the rule definition by a semicolon (`;`). All text after a semicolon until the end of a line forms the comment.

Note: The rules are introduced throughout the text of the specification when appropriate to formally define a concept. For convenience, all of the rules are gathered in the section "Field Definitions", starting on page 37.

1.2.1 Rule naming

Angle brackets (<,>) are not used, in the rule names. The name of a rule is the name itself, rather than `<name>`. Quotation marks enclose literal text, which is case sensitive. Rules are presented using the `Consolas` font family. Angle brackets are used in rule definitions, and in the rest of this document, whenever their presence will help to explain the use of rule names.

1.2.2 Alternatives: Rule1 | Rule2

Elements separated by vertical line (|) are alternatives. Therefore, "`[abc | def]`" accepts "`abc`" or "`def`".

1.2.3 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`".

1.2.4 Repetition: *Rule

The asterisk (*) character preceding an element indicates repetition. The full form is:

[1] *Section* 1.2, "Rule notation conventions" is derived from RFC 822 – Standard For The Format of ARPA Internet Messages.

`<l>*<n>element`

indicating at least `<l>` and at most `<n>` occurrences of element. Default values are 0 and infinity, 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 STAMP `FieldIdentifier`, the repeated element is represented in the data stream using the `FieldIdentifierIndex` notation, as described in Section 2 of the *STAMP Specification.*

1.2.5 Optional: [Rule]

Square brackets enclose optional elements; "`[abc def]`" is equivalent to "`*1(abc def)`".

1.2.6 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 data stream using the `FieldIdentifierIndex` notation as described in Section 2 of the *STAMP Specification*.

1.2.7 Client/Server notation convention

For the purpose of this specification, a Broadcast Feed Client is an application that establishes a connection with the Broadcast Feed Server application. The Broadcast Feed Client "listens" for output messages from the Broadcast Feed Server. The term *site* is used to identify either the Broadcast Feed Server or Client. To avoid confusion with the use of client in the business sense, *client* on its own refers to a business client of a member firm, while Broadcast Feed Client or Client (initial capital letter) refers the computer application communicating with the Broadcast Feed Server or Server at an exchange.

Chapter 2 Message Architecture

The *STAMP Specification* provides full details on message architecture. Refer to Section 2 of the STAMP document for information on data representation, message layering, control content, business content and the use of fields.

2.1 Encryption

In order to provide the confidentiality requirements within the Broadcast Feed messages (as defined by the `PrivateContent`), a form of data encryption, where the Server encrypts the data destined for a specific Client with a shared key, is used. The encryption technique used is symmetrical encryption using Data Encryption Standard (DES) in Cipher Block Chaining (CBC) mode. The following algorithm is used:

1. The fields to be encrypted (as identified in the message by belonging to the `PrivateContent` section) are gathered into a buffer, preserving the message format (i.e. `FieldChar FieldIdentifier "=" FieldValue`).
2. Prepended to the buffer is eight bytes, the first seven of which are “high-quality” random bytes, and the last byte is modulus eight of the original length of the buffer.
3. The buffer is padded (with any value) up to an eight-byte boundary.
4. The 56-bit DES key is created by taking the 64 least significant bits the MD5 hash of the shared password, and modifying the least significant bit of each byte of the 64-bit MD5 hash to represent the DES parity bit (thus reducing the 64-bit MD5 hash to a 56 bit key).
5. The CBC algorithm is applied, using the 56-bit DES key and an Initial Vector of zero, to the buffer.
6. The resultant encrypted buffer is then encoded using Base64.
7. The final `BusinessContent` of the message is then formed by the `PublicContent` of the message, `PrivateKeyIdentifier` for the destination, and `PrivateContent` equal to the Base64 result.

Sample source code for implementing this algorithm can be found in *STAMP Application Note 5 - STAMP CBC DES Algorithm.*

Chapter 3 Session Architecture

3.1 Overview

The Broadcast Feed architecture is based on the STAMP specification with one notable exception; the Broadcast Feeds are essentially “session-less”. In STAMP, communication is interactive and two-way; however, the Broadcast Feed communication is one-way from Server to Client. All subscribers to the Broadcast Feed receive the identical messages. In STAMP the client and Server negotiate parameters for message and session authentication. For the Broadcast Feed, there is no parameter negotiation.

The Broadcast Feed supports the sequencing of messages (to allow for detection and retransmission of missed messages). Retransmission is handled using the STAMP Recovery session. For more information on retransmission and recovery, refer to the *STAMP Specification*.

In summary the Broadcast Feeds are:

- Session-less
- One-way communication
- Where all parties receive the same messages

3.2 Message characteristics

Sequence numbers are assigned to each message to identify possible message sequence gaps, the Broadcast Feeds use an ascending sequence numbering scheme. Messages sent from a Server are assigned, by the Server, a sequence number which the Client then uses to determine if any messages have been missed.

If the Client detects a message gap, it should, as soon as possible, inform the Server (via the STAMP Recovery Session – see *STAMP Specification*) the sequence number range that was missed, and ask the Server to resend the messages in the range.

Implementation Note: As long as reliable transport protocols, such as TCP/IP, are used, the risk of message loss in midstream is very slight. Typically, rather than losing the occasional message in midstream, the communications path will be lost altogether, causing connection to be re-established.

Implementation Note: To help manage retransmissions, Retransmission Requests include an optional `RetransId`. When the Client asks the Server to retransmit a range of messages, it sends a unique identifier that will be included in each retransmitted message.

3.3 Session types

To support the message characteristics of the Broadcast Feed messages, a Client establishes up to three types of logical sessions with a Server:

- STAMP Recovery Session

- Equity Feed Session - all equity messages
- Equity Status Feed session - status messages for equities only

Each session type allows only a subset of the available messages. For details on allowed message types for each of the Feeds Sessions, see Section 5, Business Content Messages. For details on allowed message types for the Recovery Session, refer to *STAMP Specification*.

3.3.1 Recovery "session"

The first session a STAMP Client establishes with a STAMP Server is the Recovery Session. The purpose of a Recovery session is for a STAMP Client to request missed messages from a STAMP Server for any session at the STAMP Server the Recovery session is associated with, including the Recovery session itself. The STAMP Server requests retransmissions of STAMP Client messages via the session it detected the missed message.

Retransmitted messages are sequenced as they were in the originating session, and the session they are retransmitted for is identified by the Client via the `RetransId` field.

The Broadcast Feed Client uses the STAMP Recovery Session to request and receive retransmissions of Broadcast Feed messages.

3.3.2 Feed "session"

The Feed Session is not a true STAMP session since there is no sign-on process; rather the Broadcast Feed Client establishes a connection and begins listening for messages. Feed Sessions are used by the STAMP Client to receive the messages from the Feed Server to obtain order, trade, market state, stock state, and stock information.

There are two types of Feed Sessions where data may be received:

- Equity Feed Session
- Recon Feed Session

The STAMP Client may be required to connect to multiple Feed sessions to listen for messages about a specific stock. If multiple Feed sessions are established by TSX, all messages for a stock will be broadcast on one of the Equity Feed sessions for that trading day. For more information about the feeds, refer to section 5.11, "Equity Feed for Broadcast Feed" on page 27 and section 5.12, "Recon Feed for Broadcast Feed" on page 31.

Note: TSX reserves the right to add Equity or Recon Feed sessions with prior notice, but no migration period.

- Equity Feed - Start of day
  - Symbol Status Feed messages – Start Of Day messages to distribute detailed symbol information
  - Order Book Feed messages – Start Of Day messages to distribute "Open" order Public and Private order information

- Trading Tier Status messages – Provides statistical information to the trading community

- Equity Feed - Intraday
  - Order/Cancel Confirmations – Intraday messages for new or cancel orders
  - Change Former Order Confirmations – Intraday messages for changed orders
  - Cross Confirmations – Intraday messages for cross orders
  - Trade reports – Intraday messages for new trades
  - Trade Cancellation/Corrections – Intraday messages for corrected or cancelled trades
  - Spread Goal Notification – Intraday messages for spread goal changes
  - MOC Imbalance Notification – Intraday messages to publish the MOC Imbalance for a stock
  - General Messages – Intraday general messages
  - Market Command Response message – Intraday encrypted message response for RT Participation, Open Delay and Delay Open
  - User Notification message – Intraday encrypted message for RT Alerts
  - MBX Message – Intraday message to publish the COP and Assign Limits
  - Market State Update messages – Intraday messages to publish market state changes
  - Stock Status updates – Intraday messages to publish stock state changes
- Recon Feed (The Recon Feed will not be split at this time)
  - Order Status Feed messages – Start Of Day messages to distribute “Non-Open” order Public and Private order information
  - Anonymous Trade Feed messages – End Of Day messages to distribute Anonymous trades information

Once a Feed connection has been established, the Client process will begin receiving messages at that point in time. Any messages sent from the Broadcast Feed Server prior to the connection being established are available using the STAMP Recovery Session. For example, when the Feed connection is first established on a given day, the Broadcast Feed Client will send a RetransRequestMessage with a `LowSequenceNumber` of 1 and a `HighSequenceNumber` equal to the `SequenceNumber` less 1 of the first message received on the Broadcast Feed line. The Client will continue to receive current messages on the Broadcast Feed while receiving “old” messages on the STAMP Recovery session.

3.3.3 Addresses

The Broadcast Feed Servers and Clients exist within a STAMP network and use the Addresses defined in the STAMP implementation. The `SourceAddress` value is a unique number assigned to all STAMP clients and servers within a STAMP network. The Server side at the exchange, no matter how complex, is known to the Client by a single `STAMP-Address` per session type as described in Section 3.3, “Session types” on page 10.

The formal notation for a STAMP addresses is:

```
STAMP-Address          =   SourceAddress | DestAddress
SourceAddress          =   DirectedAddress
DestAddress            =   DirectedAddress | BroadcastAddress
DirectedAddress        =   8Hexadecimal              ; 4 bytes
BroadcastAddress       =   "00000000"                ; servers only
```

Note: Broadcast Feed servers use a `DestAddress` address of the type `BroadcastAddress`.

3.3.4 Sequence number synchronization

At the conclusion of the business day, the Broadcast Feed Server will reset its sequence numbers to “1” in preparation for the next business day.

The following algorithm describes the expected and possible scenarios for sequence numbers for messages transmitted from the Server to the Client.

```
NSSN        =  NextServerSequenceNumber  ; Server sends this to Client
LSSN        =  LastServerSequenceNumber  ; Client keeps track of this
```

Scenarios

`NSSN = LSSN + 1`

This is the ideal scenario. The Client has received every message that the Server has sent, and there is no action to take.

`NSSN > LSSN + 1`

In this scenario, the Client has missed `(NSSN - LSSN + 1)` messages sent by the Server. The Client should then send a retransmission request, through the Recovery Session, to the Server requesting the messages from `(LSSN + 1)` to `(NSSN - 1)` be retransmitted after the session has been established.

At the conclusion of the business day, the Broadcast Feed Server will reset its sequence numbers to "1" in preparation for the next business day.

The following algorithm describes the expected and possible scenarios for sequence numbers for messages transmitted from the Server to the Client.

`NSSN < LSSN, OR, NSSN = 0`

In this scenario, there are two possible problems:

- The Client has experienced a system problem whereby the Client has an incorrect idea of what messages have been received from the Server
- The Server has experienced a system problem whereby the Server has a lost or corrupted outbound message log that the Server has not been able to detect. The appropriate action is for the Server to complete the session establishment. The Client may reject any subsequent messages received from the Server with a control error report until it receives a message with sequence number equal to `LSSN+1`. The Client may also ignore any or all messages received from Server until it receives a message with sequence number equal to `LSSN+1`.

Implementation Note: The client actions noted in this section are recommendations. The client has the flexibility to both automatically resolve the gap and continue processing or to dissolve the session and investigate the gap manually.

3.4 Session dissolution

The client process simply disconnects or stops listening to messages.

3.5 Confirming connection is active

Users of STAMP will be familiar with the use of session “heartbeats” to confirm that a session with a STAMP server is still active. The Broadcast Feed does not provide any such mechanism for ensuring that the communication line is properly functioning. During periods of inactivity, the Broadcast Feed client can verify that the line is still functioning by using the retransmission control message. For example, if no messages have been received during a given period, the Broadcast Feed Client could send a `RetransRequestMessage` with a `HighSequenceNumber` equal to the last `SequenceNumber` received plus one. The Client will receive a `ControlErrorReport` (ErrorNumber 5) indicating that the requested messages are unavailable. The Control Error report will contain `LowSequenceNumber` and `HighSequenceNumber` indicating the valid range.

3.6 Heartbeat message

The HeartBeat message is a message that is used as a “Circuit assurance message” to determine whether idle time in the broadcast exceeds a specified time period.

This is a "one-way to client" message. The client only uses it for the purposes of circuit assurance.

HeartBeatMessage	=	ControlHeader [ControlTrailer]
ControlHeaderContent	=	ControlClass DestAddress HeartBeatPeriod NextSequenceNumber SourceAddress TimeStamp
Where:		
ControlClass	=	"HeartBeat"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

3.6.1 Session heartbeats

Session heartbeats are sent by the Broadcast Feed during periods of inactivity within a session to ensure the integrity of the communications path. This message is a "one-way to client" message and each message will contain a `HeartBeatPeriod`. The heartbeat message will also send the next sequence number to aid the receiver with detecting missed messages. A heartbeat message will not be sent if another message is queued for delivery.

An initial heartbeat message will be sent as the first message after successful session establishment. This will establish the `HeartBeatPeriod` for the client. A heartbeat message will be sent as soon as an inactive period (`HeartBeatPeriod` duration elapsed) has been detected by the Broadcast Feed. If during the inactive period, a heartbeat message is not received within a `HeartBeatPeriod` duration, the communication path is considered non-functional.

Chapter 4 Control Messages

Control messages are administrative messages that are primarily concerned with the mechanics of operating a STAMP session. They do not contain a `BusinessContent` section, and are therefore not sent to the trading system.

There are no control messages associated with the Broadcast Feed communication. However, retransmission requests are initiated using control messages on the STAMP Recovery session. For a complete discussion of control messages and their use refer to *STAMP Specification.*

Chapter 5 Business Content Messages

The messages described in this section are the trading messages that are broadcast from Broadcast Feed Server to the Client.

5.1 Order/Cancel Confirmation Report

The Broadcast Feed Server sends an Order/Cancel Confirmation report to the Client when a new order or cancel is entered into the trading system. An order/cancel confirmation report may also be sent to notify the Client that the attributes of an existing order have been modified.

Note: TSXV does not support `ATS-FeeCode`, `ATS-Name`, `ATS-TimeStamp`, `BasketTrade`, `MGF-Candidate`, `NonResident`, and `ProgramTrade`.

Note: TMXS does not support `Moc`, `MGF-Candidate`, `Non-Resident`, `OnStopPrice`, and `SettlementTerms`.

Note: TSX, TSXV and TMXS do not support `COMP`, `SeekDarkLiquidity, SelfTrade.`

Note: XATS does not support `Moc` and `MGF-Candidate`.

OrderCancelRespMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	PublicContent EncPrivateContent
PublicContent	=	BusinessClass BusinessAction BrokerNumber ConfirmationType OrderNumber PublicPrice Symbol TradingSysTimeStamp Volume [ATS-Name] [ATS-TimeStamp] [ByPass] [CFOdOrderNumber] [ExchangeId] [NonResident] [PriorityTimeStamp] [SettlementTerms]
PrivateContent	=	AccountType OrderDuration Price UserId [AccountId] [ActionSource] [Anonymous] [ATS-FeeCode] [BasketTrade] [CancelledReason] [COMP] [CustomerType] [Exchange-Admin] [Exchange-UserId] [ExecInst] [GatewayId] [HandInst] [IcebergRefresh] [Jitney] [MGF-Candidate] [MinQty] [Moc] [NCIB] [NoTradeFeat] [NoTradeKey] [NoTradeOrderNum] [NoTradePrice] [NoTradeVol] [OnStopPrice] [PegType] [PrivateBrokerNumber] [PrivateBusinessAction] [PrivateConfirmationType] [PrivateOrderNumber] [ProgramTrade] [ReasonCode] [RegulationId] [SeekDarkLiquidity] [SelfTrade] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TotalVolume] [Undisplayed] [UserOrderId]
Where:		
BusinessClass	=	"OrderCancelResp"
BusinessAction	=	"Buy" | "Sell"
PrivateBusinessAction	=	"Buy" | "Sell" | "Short"
ConfirmationType	=	"Accepted" | "AssignTimePriority" | "Booked" | "Cancelled" | "Frozen" | "Killed" | "Pending" |"PriceAssigned"
PrivateConfirmationType	=	"Accepted" | "AssignTimePriority" | "Booked" | "Cancelled" | "Frozen" | "Killed" | "Pending" | "PriceAssigned" |"Triggered"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.1.1 On-Stop Order/Cancel Confirmation Report

Note: TMX Select does not support this message.

An On-Stop Order/Cancel Confirmation Report is a special OrderCancelRespMessage in which all fields are encrypted (an untriggered on-stop order is a “private” order which does not enter the market until it is triggered).

The Broadcast Feed Server sends the report to the Client when a new on-stop order or a cancel on-stop order message is processed by the trading system.

The report is broadcast over the Broadcast Feed, but only the Member Firm that entered the on-stop order has access to information in the report.

Note: TSXV does not support `ATS-FeeCode`, `ATS-Name`, `ATS-TimeStamp`, `BasketTrade`, `MGF-Candidate`, and `ProgramTrade`.

Note: Undisclosed On-Stop Order is not currently supported.

OrderCancelRespMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	EncPrivateContent
PrivateContent	=	BusinessClass BusinessAction AccountType BrokerNumber ConfirmationType OnStopPrice OrderDuration OrderNumber Price Symbol TradingSysTimeStamp UserId Volume [AccountId] [ActionSource] [Anonymous] [ATS-FeeCode] [ATS-Name] [ATS-TimeStamp] [BasketTrade] [CFOdOrderNumber] [CustomerType] [ExchangeId] [Exchange-Admin] [Exchange-UserId] [GatewayId] [HandInst] [Jitney] [MGF-Candidate] [NCIB] [PriorityTimeStamp] [PrivateBrokerNumber] [PrivateConfirmationType] [PrivateOrderNumber] [ProgramTrade] [RegulationId] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TotalVolume][UserOrderId]
Where:		
BusinessClass	=	"OrderCancelResp"
BusinessAction	=	"Buy" | "Sell"
ConfirmationType	=	"Accepted" | "Cancelled"
PrivateConfirmationType	=	"Booked" | "Cancelled"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.2 Change Former (CFO) On-Stop Order Confirmation Report

Note: TMX Select does not support this message.

The Broadcast Feed Server sends a Change Former On-Stop Order Confirmation report to the Client when an untriggered On-Stop CFO is entered into the trading system.

This notification includes the details of the new order and the order number of the original untriggered order. On-Stop orders are private orders until triggered; therefore the entire business content is encrypted.

Note: TSXV does not support `ATS-FeeCode`, `ATS-Name`, `ATS-TimeStamp`, `BasketTrade`, `MGF-Candidate`, and `ProgramTrade.`

Note: Undisclosed On-Stop Order is not currently supported.

CFO-RespMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SourceAddress SequenceNumber TimeStamp [Retrans] [RetransId]
BusinessContent	=	EncPrivateContent
PrivateContent	=	AccountType BrokerNumber BusinessClass BusinessAction CFOdOrderNumber ConfirmationType NewOrderNumber OrderDuration OnStopPrice Price Symbol TradingSysTimeStamp UserId Volume [AccountId] [ActionSource] [Anonymous] [ATS-FeeCode] [ATS-Name] [ATS-TimeStamp] [BasketTrade] [CustomerType] [ExchangeId] [Exchange-Admin] [Exchange-UserId] [GatewayId] [HandInst] [Jitney] [MGF-Candidate] [NCIB] [PriorityTimeStamp] [PrivateBrokerNumber] [PrivateConfirmationType] [PrivateNewOrderNumber] [ProgramTrade] [RegulationId] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TotalVolume] [UserOrderId]
Where:		
BusinessClass	=	"CFO-Resp"
BusinessAction	=	"Buy" | "Sell"
ConfirmationType	=	"Accepted"
PrivateConfirmationType	=	"Booked"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.3 Cross Confirmation Report

The Broadcast Feed Server sends a Cross Confirmation report when an intentional cross order is entered into the trading system. This confirmation includes all relevant transaction details for both the buy side and sell side of the cross.

Note: TSXV does not support `ATS-Name`, `ATS-TimeStamp`, `BasketTrade`, `BuyATS-FeeCode`, `NonResident`, and `SellATS-FeeCode`.

Note: TMX Select does not support `Non-Resident` and `SettlementTerms`.

CrossRespMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	EncPrivateContent PublicContent
PublicContent	=	BusinessClass BusinessAction BrokerNumber ConfirmationType Symbol TradingSysTimeStamp Volume [ATS-Name] [ATS-TimeStamp] [ByPass] [CrossType] [ExchangeId] [ExtendedHours] [NonResident] [Price] [SettlementTerms]
PrivateContent	=	BuyAccountType BuyOrderNumber PrivateBusinessAction PrivateConfirmationType SellAccountType SellOrderNumber UserId [ActionSource] [Anonymous] [BasketTrade] [BuyAccountId] [BuyATS-FeeCode] [BuyCustomerType] [BuyJitney] [BuyRegulationId] [Exchange-Admin] [Exchange-UserId] [GatewayId] [HandInst] [NCIB] [PrivateBrokerNumber] [SellAccountId] [SellATS-FeeCode] [SellCustomerType] [SellJitney] [SellRegulationId] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TotalVolume] [UserOrderId]
Where:		
BusinessClass	=	"CrossResp"
BusinessAction	=	"Cross"
ConfirmationType	=	"Cross"
PrivateBusinessAction	=	"Cross" | "ShortCross"
PrivateConfirmationType	=	"Cross" | "Frozen" | "ShortCross" | “Killed”
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

Note: A Cross Confirmation Report will be followed by a Trade Report.

5.4 Trade Report

The Broadcast Feed Server sends a Trade report to the Client when a trade occurs on a previously accepted new order, change former order, 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.

Note: TSXV does not support `ATS-FeeCode`, `ATS-Name`, `ATS-SequenceNumber`, `ATS-TimeStamp`, `BasketTrade`, `MGF-Candidate`, `NonResident`, `PortfolioName`, `ProgramTrade`, and `TradeThroughExempt`.

Note: TMX Select does not support `Moc`, `Non-Resident`, `PrivateOrigPrice`, `RTAutofill`, and `SettlementTerms`.

Note: TSX, TSXV, and TMXS do not support `COMP`, `SeekDarkLiquidity`, `SelfTrade`, `MatchingPriority` and `IsDark`.

Note: XATS does not support `Moc`.

TradeReportMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	EncPrivateContent PublicContent
PublicContent	=	BusinessClass BusinessAction 2BrokerNumber 2OrderNumber Price Symbol TradeNumber TradingSysTimeStamp Volume [ATS-Name] [ATS-SequenceNumber] [ATS-TimeStamp] [1*2ByPass] [1*2CFOdOrderNumber] [CrossType] [2DisplayVolume] [ExchangeId] [ExtendedHours] [IsDark] [LastSale] [1*2Moc] [1*2NonResident] [OrigTradeID] [2PriorityTimeStamp] [SettlementTerms] [TradeCorrection] [TradeThroughExempt] [2TradeTimeStamp]
PrivateContent	=	AccountType PrivateBusinessAction RemainingVolume UserId [AccountId] [ActionSource] [1*2COMP] [1*2CustomerType] [Exchange-Admin] [1*2ExecInst] [1*2GatewayId] [1*2HandInst] [Market] [1*2MatchingPriority] [1*2MinQty] [1*2NCIB] [1*2NoTradeFeat] [1*2NoTradeKey] [2OpeningTrade] [OrderDuration] [1*2PegType] [PortfolioName] [PrincipalTrade] [PrivateBrokerNumber] [PrivateKeyIdentifier] [PrivateOrderNumber] [1*2PrivateOrigPrice] [ProgramTrade] [RegulationId] [RemainingBuyParticipationVolume] [RemainingSellParticipationVolume] [RTAutofill] [1*2SeekDarkLiquidity] [SelfTrade] [ShortMarkingExempt] [1*2SOROrderID1] [1*2SOROrderID2] [UndisclTradedVol] [1*2Undisplayed] [UserOrderId] [WashTrade]
Where:		
BusinessClass	=	"TradeReport"
BusinessAction	=	"Cancelled" | "Trade"
PrivateBusinessAction	=	"Bought" | "Sold" | "SoldShort"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

Note: 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.

Note: All Self Trades will be fully encrypted and the contents of the trade (common, sell side and buy side) will be provided on 164.0.

5.5 Stock Status Notification Message

The Broadcast Feed Server sends a Stock Status Notification to the Client when there is a change in stock status on the Trading Engine.

Note: TMX Select does not support `BlindOffsetAccepted`, `CalculatedClosingPrice`, `MGF-Volume`, `MocEligible`, `MocVwap`, `SpecialistName`, and `SpecialistPhoneNumber`.

Note: XATS does not support the following tags, `BlindOffsetAccepted`, `CalculatedClosingPrice`, `MGF-Volume`, `MocEligible`, and `MocVWAP`.

Note: TSX, TSXV, and TMXS do not support `AcceptSDL`.

Note: TSXV does not support `MGF-Volume`.

StockStatusMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass Symbol TradingSysTimeStamp [AcceptAnonymous] [AcceptSDL] [AcceptUndisplayed] [BlindOffsetAccepted] [CalculatedClosingPrice] [Comment] [Currency] [ExchangeId] [MGF-Volume] [MocEligible] [MocVWAP] [OpeningTime] [SpecialistName] [SpecialistPhoneNumber] [StockState]
Where:		
BusinessClass	=	"StockStatus"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.6 MOC Imbalance Notification Message

The Broadcast Feed Server sends a MOC Imbalance Notification to the Client when there is a MOC Imbalance for each specified stock. This message is disseminated once per stock at 3:40 p.m.

Note: TMX Select and XATS do not support this message.

MocImbalanceMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent		BusinessClass ExchangeId Symbol TradingSysTimeStamp [ImbalanceSide] [ImbalanceVolume]
BusinessClass	=	"MocImbalanceStatus"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.7 Market State Update Message

The Broadcast Feed Server sends a Market State Update to the Client when the Server receives notice of a market state change or trading session change from the Trading Engine.

MarketStateChangeMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass TradingSysTimeStamp [ExchangeId] [Exchange-Admin] [MarketState] [StockGroup]
Where:		
BusinessClass	=	"MarketStateChange"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.8 General Message

The Broadcast Feed Server sends a General message to the Client in response to a general message sent from the Trading Engine.

GeneralMessage	=	BusinessContent ControlHeader
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass MessageText TradingSysTimeStamp [BulletinIndicator] [ExchangeId] [Exchange-Admin]
Where:		
BusinessClass	=	"GeneralMessage"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.9 User Notification Message

The Broadcast Feed Server sends a User Notification message to the Client in response to an alert message sent from the Trading Engine.

Note: TSXV, TMX Select, and XATS do not support this message.

UserNotificationMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SourceAddress SequenceNumber TimeStamp [Retrans] [RetransId]
BusinessContent	=	EncPrivateContent
PrivateContent	=	BusinessClass BusinessAction Symbol TradingSysTimeStamp UserId [AccountType] [AskPrice] [BidPrice] [BrokerNumber] [BuyParticipation] [BuyParticipationVolume] [Exchange-Admin] [ExchangeId] [MessageId] [MessageText] [OrderNumber] [Price] [RemainingBuyParticipationVolume] [RemainingSellParticipationVolume] [SellParticipation] [SellParticipationVolume] [SpreadGoal] [TradeNumber] [UserOrderId] [Volume]
Where:		
BusinessClass	=	"UserNotification"
BusinessAction	=	"RTAlert"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.10 MBX Message

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 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. A generalized method for handling large messages is not a requirement for STAMP at this time; therefore fragmentation of MBX messages is handled at the application level.

Note: TMX Select does not support this message.

MBXMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SourceAddress SequenceNumber TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass BusinessAction CalculatedOpeningPrice Symbol TradingSysTimeStamp [ExchangeId] [Exchange-Admin] [MBX_PartNumber] [MBX_TotalParts] [1*OrderKey] [1*Price]
BusinessClass	=	"MBXMessage"
BusinessAction	=	"AssignCOP" | "AssignLimit"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.11 Equity Feed for Broadcast Feed

The Broadcast Feed Server sends Equity Feed messages to the Client that are sent from the Trading Engine. The Equity Feed messages consist of the following messages:

- **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
- **Symbol Status** Feed message:
  Provides information for an equity, debenture or trading instrument for the current trading day.
  - The Symbol Status Feed message contains only public STAMP message fields.
  - This feed is available at the beginning of each trading day.
  - The `LastMessage` STAMP tag indicates the completion for the Symbol Status STAMP messages.
- **Order Book** Feed message (TSX, TSXV, and XATS):
  Provides both the public and private (encrypted) information for all open orders in the market. The information is as of the end of the previous business day. The information is used by client gateways to maintain market information locally.
  - The Order Book Feed messages contain only public and private (encrypted) STAMP message fields for open orders only.
  - This feed is available at the beginning of each trading day.
  - The LastMessage STAMP tag indicates the completion for the Order Book STAMP messages.

5.11.1 Trading Tier Status Message

The Broadcast Feed Server sends **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

TradingTierStatusMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent	=	BusinessClass BusinessAction ExchangeId TotalNumOpenOrders TotalNumStockGroups TotalNumSymbols TradingSysTimeStamp TradingTierId
Where:		
BusinessClass	=	"MarketInfo"
BusinessAction	=	"TradingTierStatus"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.11.2 Symbol Status Feed Message

The Broadcast Feed Server sends Symbol Status Feed messages to the Client that are sent from the Trading Engine.

Symbol Status Feed messages provide information for an equity, debenture or trading instrument for the current trading day.

Note: TMX Select does not support `MGF-Setting`, `MGF-Volume`, `MocEligible`, `SpecialistName`, `SpecialistPhoneNumber`.

Note: XATS does not support `MGF-Setting`, `MGF-Volume`, `MocEligible`

Note: TSX, TSXV, and TMXS do not support `AcceptSDL`.

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

5.11.3 Order Book Feed Message

Note: TMX Select does not support this message.

The Broadcast Feed Server sends Order Book Feed messages to the Client that are sent from the Trading Engine.

Order Book Feed messages provide the public and private information for all open orders in the market. The information is as of the end of the previous business day. The information is used by client gateways to maintain market information locally (TSX, TSXV and XATS).

Note: TSXV does not support `BasketTrade`, `MGF-Candidate`, `NonResident`, or `ProgramTrade`.

Note: XATS does not support `MGF-Candidate`.

OrderBookMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent		[PublicContent] EncPrivateContent
PublicContent	=	BusinessClass BusinessAction BrokerNumber ExchangeId Volume MarketSide OrderNumber PublicPrice Symbol TradingSysTimeStamp [LastMessage] [NonResident] [NumberOfMessages] [PriorityTimeStamp] [SettlementTerms] [TotalNumMessages]
PrivateContent	=	AccountType OrderAction OrderDuration OrderStatus Price RemainingVolume UserId [AccountId] [ActionSource] [Anonymous] [BasketTrade] [CustomerType] [ExecInst] [GatewayId] [HandInst] [Jitney] [MGF-Candidate] [MinQty] [NCIB] [NoTradeFeat] [NoTradeKey] [OnStopPrice] [PegType] [PrivateBrokerNumber] [PrivateOrderNumber] [ProgramTrade] [RegulationId] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TotalVolume] [Undisplayed] [UserOrderId]
Where:		
BusinessClass	=	"OrderInfo"
BusinessAction	=	"OrderBook"
Allowed SessionType		
SessionType	=	"EquityFeed" | "Recovery"

5.12 Recon Feed for Broadcast Feed

The Broadcast Feed Server sends Recon Feed messages to the Client that are sent from the Trading Engine. The Recon Feed messages consist of the following messages:

- **Order Status** Feed message:
  Provides both the public and private (encrypted) information for all non-open orders in the market. It is primarily used as a reconciliation mechanism by which to verify previous days' trading status.
  - The Order Status Feed message will contain both public and private (encrypted) STAMP message fields.
  - This feed is available at the beginning of each trading day.
  - The LastMessage STAMP tag indicates the completion for the Order Status STAMP messages.
- **Anonymous Trades** Feed message:
  Provides return trade information for all anonymous trades that occurred during the day for the member firm.
  - The Anonymous Trades Feed message contains both public and private (encrypted) STAMP message fields.
  - This feed is available at the end of the business day.
  - The LastMessage STAMP tag indicates the completion for the Anonymous Trades STAMP messages.

Note: The Recon Feed also carries copies of the following Equity Feed messages:

- Trading Tier Status message (see page 28)
- Symbol Status Feed message (see page 29)
- Order Book Feed message (see page 30)

5.12.1 Order Status Feed Message

The Broadcast Feed Server sends Order Status feed messages to the Client sent from the trading engine.

An Order Status Feed message provides the current status of a given non-open order. It is primarily used as a reconciliation mechanism by which to verify previous days' trading status. The Order Status Feed Message will contain both public and private (encrypted) STAMP message fields. This feed is available at the beginning of each trading day.

Note: TSXV does not support `BasketTrade`, `MGF-Candidate`, `NonResident` or `ProgramTrade.`

Note: TMX Select does not support `MGF-Candidate`, `Moc`, `OnStopPrice`, and `SettlementTerms`.

Note: TSX, TSXV and TMXS do not support `COMP`, and `SeekDarkLiquidity`.

Note: XATS does not support `MGF-Candidate`, and `Moc`.

OrderStatusMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent		PublicContent EncPrivateContent
PublicContent	=	BusinessClass BusinessAction ExchangeId [ByPass] [NumberOfMessages] [TotalNumMessages]
PrivateContent	=	AccountType OrderAction OrderDuration OrderNumber OrderStatus Price Symbol TradingSysTimeStamp UserId Volume [AccountId] [ActionSource] [Anonymous] [BasketTrade] [BrokerNumber] [COMP] [CustomerType] [DisplayVolume] [ExecInst] [GatewayId] [HandInst] [Jitney] [LastMessage] [MGF-Candidate] [MinQty] [Moc] [NCIB] [NonResident] [NoTradeFeat] [NoTradeKey] [OnStopPrice] [PegType] [PriorityTimeStamp] [ProgramTrade] [ReasonCode] [RegulationId] [RemainingVolume] [SeekDarkLiquidity] [SettlementTerms] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [Undisplayed] [UserOrderId]
Where:		
BusinessClass	=	"OrderInfo"
BusinessAction	=	"OrderStatus"
Allowed SessionType		
SessionType	=	"ReconFeed" | "Recovery"

5.12.2 Anonymous Trades Feed Message

The Broadcast Feed Server sends Anonymous Trades Feed messages to the Client that are sent from the Trading Engine.

The Anonymous Trades Feed message provides trade information for all anonymous trades that occurred during the day for the member firm. The Anonymous Trades Feed Message contains both public and private (encrypted) STAMP message fields.

This Feed is available at the end of the business day (TSX, TSXV, TMXS, and XATS).

Note: TSXV does not support `ATS-Name`, `BasketTrade`, `NonResident`, and `ProgramTrade.`

Note: TMX Select does not support `Moc` and `RTAutofill`.

Note: TSX, TSXV, and TMXS do not support `COMP`, `SeekDarkLiquidity`, `SelfTrade`, `MatchingPriority` and `IsDark`.

Note: XATS does not support `Moc`.

AnonymousTradesMessage	=	ControlHeader BusinessContent
ControlHeaderContent	=	DestAddress SequenceNumber SourceAddress TimeStamp [Retrans] [RetransId]
BusinessContent		PublicContent EncPrivateContent
PublicContent	=	BusinessClass BusinessAction ExchangeId [ByPass] [IsDark]
PrivateContent	=	AccountType BrokerNumber OppositeBroker OrderNumber Price PrivateBusinessAction Symbol TradeNumber TradingSysTimeStamp UserId Volume [AccountId] [ActionSource] [Anonymous] [ATS-Name] [BasketTrade] [COMP] [CustomerType] [ExecInst] [GatewayId] [HandInst] [Jitney] [LastMessage] [Market] [MatchingPriority] [MinQty] [Moc] [NCIB]] [NoTradeFeat] [NoTradeKey] [NumberOfMessages] [OpeningTrade] [PegType] [PrincipalTrade] [ProgramTrade] [RTAutofill] [SeekDarkLiquidity] [SelfTrade] [ShortMarkingExempt] [SOROrderID1] [SOROrderID2] [TradeCorrection] [TradeDate] [TradeTimeStamp] [Undisplayed] [UserOrderId] [WashTrade]
Where:		
BusinessClass	=	"TradeInfo"
BusinessAction	=	"AnonymousTrades"
Allowed SessionType		
SessionType	=	"ReconFeed" | "Recovery"

Chapter 6 Multicast Transport Layer Specification

This section describes the Multicast Transport Layer Specification (TLS) for the TSX Broadcast Feed (TBF), TSXV Broadcast Feed (CBF), TMX Select Broadcast Feed (SBF), and Alpha Broadcast Feed (ABF).

This Transport Layer Specification (TLS) has been designed using the following key principles:

- Multicast is not a reliable service.
- The message framing format will be based on that of the STAMP TLS[2], in order to minimize the impact of the multicast change.

The TLS will support message fragmentation and reassembly.

For more information about receiving the Broadcast Feeds over multicast, refer to *TMX Equity Market Multicast Service Access Guide*.

6.1 Formal message definition

The formal message definition of the Multicast Transport Layer for TBF, CBF, and SBF is:

```
TransportInterfaceSection = TransportHeader TransportTrailer
TransportHeader           = TLSH-Element | XTLSH-Element3
TransportTrailer          = TLST-Element

TLSH-Element = TransportHeaderChar LengthOfLength Length
TLST-Element = TransportTrailerChar

TransportHeaderChar  = <US-ASCII STX; Start of Text>  ; 0x02
TransportTrailerChar = <US-ASCII ETX; End of Text>    ; 0x03

LengthOfLength = 1Digit                ; the size of the Length field
                                       ; (bytes)

Length         = <LengthOfLength>Digit ; size of ControlSection plus
                                       ; BusinessContentSection in
                                       ; bytes
                                       ; (variable number of digits)
```

[2] For details, please refer *STAMP Transport Layer Specification - TCP/IP*.
[3] This transport header, which provides for message fragmentation, will be defined in the future.

This definition provides two types of message framing:

- **The Standard Framing Format** (TransportHeader = TLSH-Element)
  When using this format, the assumption is that each feed message is smaller than the Maximum Transmission Unit (MTU) for Ethernet (i.e., 1500 bytes, including all headers). Though this assumption is currently valid[4], feed messages are anticipated in the future to exceed the MTU.
- **The Extended Framing Format** (TransportHeader = XTLSH-Element)

This format, which supports the fragmentation of feed messages exceeding the MTU, will be available in the future.

6.2 Message examples

In the following examples:

- THC `= TransportHeaderChar`
- LOL `= LengthOfLength`
- L = `Length`
- Payload `= ControlSection BusinessContentSection`
- TTC `= TransportTrailerChar`

[4] The current maximum length of feed messages is lower than 1000 bytes.

6.2.1 Standard framing format

THC	LOL	L	L	L	*Payload*	TTC
STX	3	1	1	0	*(110 bytes of data)*	ETX

THC	LOL	L	L	L	*Payload*	TTC
STX	3	5	9	3	*(593 bytes of data)*	ETX

THC	LOL	L	L	L	*Payload*	TTC
STX	3	0	2	5	*(25 bytes of data)*	ETX

THC	LOL	L	L	*Payload*	TTC
STX	2	6	5	*(65 bytes of data)*	ETX

THC	LOL	TTC
STX	0	ETX

Note: There is no payload for this message.

6.2.2 Extended framing format

Note: This format is not yet supported.

Chapter 7 Field Definitions

Note: Field Identifier tags universal to TSX, TSXV, TMXS, and XATS are not marked with an exchange identifier.

A

AcceptAnonymous

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

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

AcceptSDL

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

```
FieldIdentifier = 622   ; XATS only
AcceptSDL = “Y” | “N”   ; default is “Y”
```

AcceptUndisplayed

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

```
FieldIdentifier         = 605 ; Default = “Y”
AcceptUndisplayed       = “Y” | “N”
```

AccountId

Identifies the trading account identification.

```
FieldIdentifier = 1
AccountId       = 1*15AlphaNumeric  ; no default
```

AccountType

Type of the trading account.

```
FieldIdentifier = 2
AccountType     = “CL”  |   ; client
                  “IN”  |   ; inventory
                  “MP”  |   ; ME pro order entered on TSX, TMXS and XATS
                  “NC”  |   ; non-client ,default
                  “OT”  |   ; options market maker
                  “OF”  |   ; options firm account, TSX, TMXS and XATS
                  “ST”  |   ; equities specialist, TSX, TMXS and XATS
```

ActionSource

Source of the action performed on an order.

```
FieldIdentifier = 75
ActionSource    = 1*3AlphaNumeric
```

Note: For TSX, TSXV, TMXS and XATS, `ActionSource` is an optional 3-character field used by the vendors. Each vendor will be supplied with this value by the TSX.

Note: If no `ActionSource` is sent, TSX/TSXV/TMXS/XATS will send the default value that was assigned.

AlphaNumeric

Alphabetic and numeric characters.

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

Anonymous

Indicates an order is an anonymous order.

```
FieldIdentifier = 129
Anonymous       = "Y" | "N"  ; default is "N"
```

AskPrice

The price a seller is willing to accept.

```
FieldIdentifier = 180
AskPrice        = Price
```

ATS-FeeCode

The billing tier code associated with the Alternative Trading System.

```
FieldIdentifier = 391              ; TSX and TMXS
ATS-FeeCode     = 1*3AlphaNumeric  ; no default
```

ATS-Name

The Alternative Trading System where the transaction originated.

```
FieldIdentifier = 351              ; TSX and TMXS
ATS-Name        = 1*3AlphaNumeric  ; no default
```

Note: ATS-Name is an optional 3-character field used by the vendors. Each vendor will be supplied this value by the TSX.

ATS-SequenceNumber

The message sequence number from the Broadcast Feed message, which represents the last Continuous Book update used by the ATS.

```
FieldIdentifier    = 352           ; TSX and TMXS, no default
ATS-SequenceNumber = 1*9Digit      ; 0 to 999 999 999
```

ATS-TimeStamp

The time of the transaction at the originating ATS. This will be supplied by the ATS.

```
FieldIdentifier = 350              ; TSX and TMXS
ATS-TimeStamp   = TradeTimeStamp
```

B

Base64

Encoded binary representation.

```
Base64 = Digit | <US-ASCII "A" to "Z", "a" to "z", "+", "/", or "=">
         ; 0x30 to 0x39, 0x3d, 0x41 to 0x5a, 0x61 to 0x7a
```

BasketNumber

The number assigned to a basket trade.

```
BasketNumber = 1*5Digit     ; TSX, TMXS and XATS
```

BasketTrade

The order is part of a basket trade.

```
FieldIdentifier = 42                  ; TSX, TMXS and XATS
BasketTrade     = BasketNumber | "N"  ; default is "N"
                                        to indicate a program trade
```

BidPrice

The price a buyer is willing to pay.

```
FieldIdentifier = 179
BidPrice        = Price
```

BlindOffsetAccepted

Value identifying that MOC Blind Offsetting orders have been accepted.

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

BoardLot

Board lot volume.

```
FieldIdentifier = 115
BoardLot        = Volume
```

BroadcastAddress

A STAMP-Address that means all other addresses in a STAMP network.

```
BroadcastAddress = "00000000"
```

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  = "AnonymousTrades"  |
                  “AssignCOP”        | ; TSX, TSXV and XATS
                  “AssignLimit”      | ; TSX, TSXV and XATS
                  “Buy”              |
                  “Cancelled”        |
                  “Cross”            |
                  “DelayOpenStock”   | ; TSX only
                  ”OpenDelayedStock” | ; TSX only
                  "OrderBook"        | ; TSX, TSXV and XATS
                  "OrderStatus"      | ; TSX, TSXV and XATS
                  “Participation”    | ; TSX only
                  “RTAlert”          | ; TSX only
                  “Sell”             |
                  "SymbolStatus"     |
                  “Trade”            |
                  “TradingTierStatus”|
```

BusinessClass

The message class for a Business Content Layer message.

```
FieldIdentifier = 6  ; no default, maximum 35 Characters
BusinessClass   = “CFO-Resp”                |
                  “CrossResp”               |
                  “GeneralMessage”          |
                  “MarketCommandResp”       |
                  “MarketInfo”              |
                  “MarketStateChange”       |
                  “MBXMessage”              |  ; TSX, TSXV and XATS
                  “MocImbalanceStatus”      |  ; TSX and TSXV
                  “OrderCancelResp”         |
                  “OrderInfo”               |  ; TSX, TSXV and XATS
                  “StockStatus”             |
                  “SymbolInfo”              |
                  “TradeInfo”               |
                  “TradeReport”             |
                  “UserNotification”           ; TSX and TSXV
```

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.0

BusinessContentSection

The BusinessContent and BusinessContentChar.

```
BusinessContentSection = Empty | BusinessContent
```

BuyAccountId

The buyer's trading account identification.

```
FieldIdentifier = 7
BuyAccountId    = AccountId
```

BuyAccountType

The buyer's type of trading account.

```
FieldIdentifier = 8
BuyAccountType  = AccountType
```

BuyATS-FeeCode

Marker to indicate the buy side of the billing tier code associated with the Alternative Trading System.

```
FieldIdentifier = 395             ; TSX and TMXS
BuyATS-FeeCode  = 1*3AlphaNumeric
```

BuyCustomerType

Identifies the cross buy side customer account type.

```
FieldIdentifier = 509
BuyCustomerType = 3Digit   ; 000 to 999, (no default)
```

BuyJitney

Marker to indicate that the buy side of a cross is being executed on behalf of another broker.

```
FieldIdentifier = 169
BuyJitney       = Jitney
```

BuyOrderNumber

The trading-system-assigned number for the buyer's order.

```
FieldIdentifier = 10
BuyOrderNumber  = OrderNumber
```

BuyParticipation

To indicate if the responsible Equities Specialist's participation on the buy side is active.

```
FieldIdentifier   = 167           ; TSX only
BuyParticipation  = "On" | "Off"  ; no default
```

BuyParticipationVolume

To assign the maximum buy participation volume for a symbol.

```
FieldIdentifier        = 592     ; TSX only
BuyParticipationVolume = Volume  ; no default
```

BuyRegulationId

Identification marker for UMIR specific designations to orders and trades.

```
FieldIdentifier = 326
BuyRegulationId = RegulationId
```

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. It is published only when outside acceptable limits set by the Exchange, triggering an extension to the MOC limit order entry session.

```
FieldIdentifier        = 491    ; TSX and TSXV
CalculatedClosingPrice = Price
```

CalculatedOpeningPrice

The price at which orders will trade at the opening.

```
FieldIdentifier        = 191   ; TSX, TSXV and XATS
CalculatedOpeningPrice = Price
```

CancelledReason

Indicates that the order has been cancelled because of Cancel on Disconnect (COD).

```
FieldIdentifier = 608
CancelledReason = 1    ; Cancelled because of COD
```

CFOdOrderNumber

The original order number of the order that was CFOd.

```
FieldIdentifier = 11
CFOdOrderNumber = OrderNumber
```

Comment

A text field corresponding to a reason code entered by Equity Operations when a stock is halted, when the initiator delays opening on a stock, or when there is a change to the RT/Odd Lot Trader on a stock. It is also a system-generated text field describing the disabling of a MOC session by TSX Trading Services.

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

Notes for RT (TSX):

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

Notes for Odd Lot Trader (TSXV and XATS):

- When an Odd Lot Trader is added to a stock (from no Odd Lot Trader to having an Odd Lot Trader) – Comment = “OLT Change”
- When an Odd Lot Trader exists for a stock and is changed – Comment = “OLT Change”
- When an Odd Lot Trader is removed from a stock (leaving no Odd Lot Trader) – Comment = “OLT Removed”

Notes for MOC-eligible stock (TSX and TSXV):

MOC Disabled, See Trader Notes for Details

COMP

Contra Order Matching Preference. A dark order attribute indicating the types of orders it will trade against.

```
FieldIdentifier  = 612                  ; XATS only
COMP =             "1"                | ; SDL (default)
                   "2"                | ; Dark
                   "3"                | ; All
```

ConfirmationType

The type of confirmation for a report.

```
FieldIdentifier  = 16
ConfirmationType = "Accepted"               |
                   "AssignTimePriority"     |
                   "Booked"                 |
                   "Cancelled"              |
                   "Cross"                  |
                   "Frozen"                 |
                   "Killed"                 |
                   "Pending"                |   ; TSX, TSXV, and XATS
                   "PriceAssigned"
```

ControlClass

The message class for a Control Layer message.

```
FieldIdentifier = 3           ; no default
ControlClass    = "HeartBeat"
```

ControlContent

The ControlHeader and ControlHeaderChar.

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.

ControlSection

The ControlHeader and ControlTrailer section of a STAMP message.

```
ControlSection = ControlHeader [ControlTrailer]
```

CrossType

Refers to one of five specialty cross types including Basis, VWAP, Contingent, Internal, and STS crosses. These crosses are treated differently from regular crosses with respect to interference and or price validation.

```
FieldIdentifier = 390        ; no default
CrossType       = "Basis"  |; Basis, TSX, TMXS and XATS
                  "Contgt" |; Contingent, TSX, TMXS and XATS
                  "Intrnl" |; Internal
                  "STS"    |; Special Trading Session, TSX and XATS
                  "VWAP"    ; Volume Weighted Average Price, TSX, TMXS and XATS
```

Currency

The currency of a price.

```
FieldIdentifier = 58       ; no default
Currency        = "CAD"  | ; Canadian dollars
                  "USD"    ; US dollars
```

CUSIP

Clearing and settlement registration number.

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

CustomerType

Identifies the customer account type.

```
FieldIdentifier         = 504
CustomerType            = 3Digit  ; 000 to 999, (no default)
```

D

Date

The date format.

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

DESPrivateContent

The private business fields contained in a message.

```
FieldIdentifier    = 164   ;
DESPrivateContent  = PrivateContent ; DES encrypted, Base64 encoded
```

Note: An encrypted blob that, when decrypted, contains the Fields defined in the `PrivateContent` of the message.

DestAddress

The destination STAMP address.

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

Note: 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)
```

Note: The value is a 4-byte value encoded in 8-byte hexadecimal format.

DisplayVolume

Public remaining volume.

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

E

Empty

Nothing.

```
Empty = ""
```

EncPrivateContent

The elements relating to the encrypted part of a message.

```
EncPrivateContent = PrivateKeyIdentifier
                    DESPrivateContent  ; Encrypted Fields
```

Note: `EncPrivateContent` consists of a key and an encrypted blob that when decrypted contains `1*Field`.

ExchangeId

Identifies the exchange from which the message originated.

```
FieldIdentifier = 247
ExchangeId      = "CDX"  | ; TSXV only
                  "TSE"  | ; TSX only
                  "SEL"    ; TMXS only
                  "ALM"  | ; XATS only
```

Exchange-Admin:

An assigned marker to transmit information to the broadcast feed and order entry sessions.

```
FieldIdentifier = 380
Exchange-Admin  = 0*36AlphaNumeric
```

Note: The Exchange-Admin tag has 36 possible markers of which 8 are currently defined. Any one of these markers that are described as "Exchange Use Only" should be ignored by the vendors. Any marker with a value of zero (0) should also be ignored by the vendors. Position 5 and 6 are applicable only to orders sent to the TSX Smart Order Router (SOR).

The following table defines how this tag is used. Currently there are 8 positions used with 28 available.

Position	Marker Description	Marker Value
0	Trade Source	'T' - TSX 'X' - TSXV 'S' – TMXS 'A' - XATS
1	Order Classification	'A' – Active 'P' - Passive
2	Trading Session	'O' – Opening (TSX, TSXV, and XATS) 'P' - Post Open 'E' - Last Sale Trading Session (TSX, TSXV, and XATS)
3	Responsible RT/Oddlot Trader for Stock	'Y' - UserId is the RT/Oddlot Trader for the stock and is the buyer and/or the seller (TSX, TSXV, and XATS) 'N' - UserId is not the RT/Oddlot Trader of the stock (TSX, TSXV, and XATS) '0' - Position 3 should be ignored (TMXS)
4	Executed Against Dark Liquidity	'D' – Order executed against dark liquidity
5	Inbound Routing Instructions	'N' (SOR Software pass-through only - No routing) 'X' Directed TSX 'V' Directed TSXV 'L' Directed TMXS 'P' Directed Pure 'C' Directed Chi-X 'O' Directed Omega 'A' Directed Alpha '2' Directed CX2 'Y' Directed Lynx
6	Routing Confirmation	'X' TSX 'V' TSXV 'L' TMXS 'P' Pure 'C' Chi-X 'O' Omega 'A' Alpha '2' CX2 'Y' Lynx

7	Execution State	'A' Delayed Active
8 to 35	Not Yet Defined	

Exchange-UserId

The user id for an exchange staff member (e.g., Trading Service Representative)

```
FieldIdentifier = 265
Exchange-UserId = UserId
```

ExDestination

The execution destination as defined by the institution when the order is entered.

```
FieldIdentifier = 607
ExDestination   = TMXS      ; TMX Select
                = XATS      ; Alpha
```

ExecInst

Instructions for order handling. If more than one instruction is applicable on an order, this field can contain multiple instructions, each separated by a space.

```
FieldIdentifier = 591
ExecInst        = 1*7Alphanumeric ; no default
                  "0"           | ; Stay on offer side
                  "9"             ; Stay on bid side
```

ExtendedHours

To indicate action occurred during extended hours session, now known as the Last Sale Trading Session.

```
FieldIdentifier = 76         ; TSX, TSXV and XATS
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
```

G

GatewayId

STAMP Address that identifies the gateway to which the transaction was routed.

```
FieldIdentifier = 311
GatewayId       = DirectedAddress
```

H

HandInst

Handling instruction for trade and quote order protection.

```
FieldIdentifier = 596
HandInst        = 1 | ; Default. Directed Action Order (DAO): The Exchange
                      ; does not prevent trade-throughs or locked/crossed
                      ; markets.
                  5 | ; Kill/Cancel: The Exchange will kill or cancel any
                      ; portion of the incoming order that would trade
                      ; through or lock/cross an away market.
                  6   ; Reprice: The Exchange will prevent trade-throughs
                      ; and locked/crossed markets by adjusting the price and
                      ; booking the order; once booked, it remains static at
                      ; that price.
```

HeartBeatPeriod

The period of time between HeartBeat messages.

```
FieldIdentifier = 23
HeartBeatPeriod = 1*4Digit  ; 1 to 3600 seconds (default is 30)
```

Hexadecimal

Hexadecimal number representation.

```
Hexadecimal = Digit | “a” | “b” | “c” | “d” | “e” | “f”
```

I

IcebergRefresh

An indicator as to whether the order is a result of an iceberg’s display volume being refreshed.

```
FieldIdentifier = 606        ;
IcebergRefresh  = "Y" | "N"  ; default is "N"
```

ImbalanceSide

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

```
FieldIdentifier = 492              ; no default, TSX only
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           ; TSX Only
ImbalanceVolume = 1*9 digit     ; no default
```

IsDark

Indicates at least one side of the trade is IntraSpread dark.

```
FieldIdentifier = 617           ; XATS Only
IsDark          = "Y" | "N"  ;
```

J

Jitney

An order is marked as being executed on behalf of another broker.

```
FieldIdentifier = 25
Jitney          = BrokerNumber | “N”  ; default is “N”
```

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:

- **Symbol Status message**: Set the `LastMessage` indicator to ‘Y’ on the last symbol message in the stock group
- **Order Book message (TSX, TSXV and XATS)**: Set the `LastMessage` indicator to ‘Y’ on the last open order message in the stock group
- **Order Status message (TSX, TSXV and XATS)**: Set the `LastMessage` indicator to ‘Y’ on the last removed order message in the stock group
- **Anonymous Trades message**: Set the LastMessage indicator to ‘Y’ on the last anonymous trade message in the stock group

```
FieldIdentifier = 113
LastMessage     = "Y" | "N"  ; default is "N"
```

LastSale

Last sale price of a stock.

```
FieldIdentifier = 114
LastSale        = NumericPrice
```

ListingMkt

The market on which a security is listed.

```
FieldIdentifier = 554
ListingMkt      = “TSE” |   ; for TSX
                = “CDX” |   ; for TSXV
                = “ALM”     ; for Alpha
```

LowSequenceNumber

The lowest sequence number asked for in a Retransmission Request.

```
FieldIdentifier   = 27
LowSequenceNumber = SequenceNumber
```

M

Market

Indicates the market at which a trade has occurred.

```
FieldIdentifier = 283
Market          = “Boardlot”       | ; no default
                  “Oddlot”         | ; TSX, TSXV and XATS
                  “SpecialTerms”     ; TSX, TSXV and XATS
```

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     = “CCP Determination”        |    ; TSX and TSXV
                  “Closing”                  |    ; TSX and TSXV
                  “Closed”                   |
                  “Extended Hours Open”      |    ; TSX, TSXV and XATS
                  “Extended Hours Close”     |    ; TSX, TSXV and XATS
                  “Extended Hours CXLs”      |    ; TSX, TSXV and XATS
                  “MOC Imbalance”            |    ; TSX and TSXV
                  “Open”                     |
                  “Opening”                  |    ; TSX, TSXV and XATS
                  “Pre-open”                 |    ; TSX, TSXV and XATS
                  “PriceMovementExtension”       ; TSX and TSXV
```

MarketType

The type of market the parameters refer to.

```
FieldIdentifier = 152         ; no default
MarketType      = "Equity"
```

MatchingPriority

Indicates the type of priority used to match the order in a trade.

```
FieldIdentifier   = 615     ; XATS only
MatchingPriority  = "1"  |  ; Broker Indicates match was due to
                              Broker Preferencing
```

MaxMessageLength

The maximum length of a STAMP message.

```
MaxMessageLength = 4096  ; the sum of the length of an entire STAMP
                         ; message (in bytes)
```

MBX_PartNumber

Number identifying an MBX part message.

```
FieldIdentifier = 194       ; TSX, TSXV and XATS
MBX_PartNumber  = 1*3Digit  ; no default
```

MBX_TotalParts

Total number of parts in a fragmented MBX message.

```
FieldIdentifier = 195       ; TSX, TSXV and XATS
MBX_TotalParts  = 1*3Digit  ; no default
```

MessageId

A unique identifier assigned by the STAMP Client to a Business Content message that is not an order.

```
FieldIdentifier = 106
MessageId       = 1*16AlphaNumeric  ; no default
```

MessageText

The description for a trading system generated message.

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

MGF-Candidate

A marker as to whether an order is eligible for Minimum Guaranteed Fill.

```
FieldIdentifier = 30         ; TSX only
MGF-Candidate   = “Y” | “N”  ; default is “N”
```

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      ; TSX only
MGF-Volume      = Volume
```

MinQty

The minimum volume of an order that is required for a trade.

```
FieldIdentifier = 598
MinQty          = Volume        ; no default
```

Note: For XATS, this field is allowed only when COMP is set to “Dark” or “All”.

Moc

Identifies the order as a Market On Close order.

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

MocEligible

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

```
FieldIdentifier = 496          ; TSX and TSXV
MocEligible     = "Y" | "N"    ; no default
```

MocVwap

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

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

N

NCIB

Identifies Normal-Course Issuer Bid (NCIB) orders; the action of a company buying back its own outstanding shares from the markets so it can cancel them.

```
FieldIdentifier  = 505
NCIB             = "Y" | "N"  ; default "N"
```

NewOrderNumber

The new trading system assigned order number after a CFO.

```
FieldIdentifier = 32
NewOrderNumber  = OrderNumber
```

NonResident

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

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

NoTradeFeat

A marker that is supplied by the Member Firm to either

- prevent trading against that same Member Firm's contra orders based on a matching `NoTradeKey`, or
- accept the trade with that same Member Firm's contra order based on matching `NoTradeKey` and suppress the execution from the public feed.

```
FieldIdentifier  = 586       ;
NoTradeFeat = 2*AlphaNumeric ; No default, valid values:
                             ; 1st char "N"=Cancel newest, or
                                        "E"=Execute match (valid only for XATS)
                             ; 2nd char "M"=Prevent trade at NoTradeKey level
```

NoTradeKey

A Member Firm generates these keys to

- prevent trading against that same Member Firm's contra orders based on a matching NoTradeKey, or
- accept the trade with that same Member Firm's contra order based on matching NoTradeKey and suppress the execution from the public feed.

Note that the marketplace does not generate this key or enforce the uniqueness of this key. NoTradeKey only prevents trades between orders generated by the same BrokerNumber (or if present, by PrivateBrokerNumber).

```
FieldIdentifier  = 587             ;
NoTradeKey       = 1*6Alphanumeric ; No default
```

NoTradeOrderNum

The private resting order number that would have matched with the active order, if not prevented by the no-trade feature..

```
FieldIdentifier  = 588
NoTradeOrderNum  = OrderNumber     ; No default
```

NoTradePrice

The price the match would have occurred at, if not prevented by the no-trade feature.

```
FieldIdentifier  = 589
NoTradePrice     = Price           ; No default
```

NoTradeVol

The number of shares that would have matched, if not-prevented by the no trade feature.

```
FieldIdentifier  = 590
NoTradeVol       = Volume          ; No default
```

NumberOfMessages:

Number of messages contained in the query response. It varies, depending on the type of message:

- **Symbol Status message**: Assign incremental numbers for each symbol message in the stock group
- **Order Book message (TSX, TSXV and XATS)**: Assign incremental numbers for each open order message in the stock group
- **Order Status message (TSX, TSXV and XATS)**: Assign incremental numbers for each removed order message in the stock group
- **Anonymous Trades message**: Assign incremental numbers for each anonymous trade message in the stock group

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

Numeric

Digits from 0 to 9.

```
Numeric = all US-ASCII character; 0x00 to 0x09
```

NumericPrice

A price in a currency.

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

O

OnStopPrice

Trigger the order when the stock trades at or through the OnStopPrice.

```
FieldIdentifier = 35           ; TSX, TSXV and XATS
OnStopPrice     = NumericPrice
```

OpeningTime

The opening time for a stock.

```
FieldIdentifier = 120
OpeningTime     = Time  ; default is value of OpenEntry
```

OpeningTrade

An indicator as to whether the trade occurred at the opening or was the first board lot trade of the day.

```
FieldIdentifier = 274
OpeningTrade    = "Y" | "N"  ; default is "N"
```

OppositeBroker

The broker with whom the trade occurred.

```
FieldIdentifier = 37
OppositeBroker  = BrokerNumber
```

OrderAction

The BusinessAction for the order.

```
FieldIdentifier = 99             ; no default
OrderAction     = "Buy"     |
                  "Sell"    |
                  "Short"   |
                  "Cross"
```

OrderDuration

The time for which a trade is valid.

```
FieldIdentifier = 39              ; default is "Day"
OrderDuration   = "Day"          |
                   Date          | ; XATS MX, TSX and TSXV
                  "FOK"          | ; XATS, TSX, TSXV, and TMXS
                  "GTC"          | ; XATS, MX, TSX and TSXV
                  "IOC"          | ; XATS, TSX, TSXV, and TMXS
                  "WC"           | ; MX only
                  "GFD"          | ; XATS
```

Definitions:

Day: Will automatically be cancelled at the end of the same day it was entered if it has not received a complete fill.

Date: Will remain in the system until it is filled or until the date specified at which time it is automatically killed by the system. Note: For XATS, TSXV and TSX Date orders are only valid for 90 days. For MX, Date orders are supported until expiry of instrument.

FOK: Will fill all or kill all.

GTC: Will remain in the system until filled for up to a maximum of one year. Note: For XATS, TSXV and TSX, GTC orders are only valid for 90 days. For MX, GTC orders are supported until expiry of instrument.

IOC: Will trade as much stock upon entry to the system, but will immediately kill any unfilled volume.

WC: A Day order with WhileConnected functionality. Day orders that are valid for a day and/or are purged from the system in case of disconnection of the participant.

GFD: Good for post open trading session only. For XATS only.

OrderKey

Unique key identifying orders in the system.

```
FieldIdentifier = 192     ; TSX, TSXV and XATS
OrderKey        = BrokerNumber "|" OrderNumber  ; no default
```

OrderNumber

A number assigned to the order by the trading system.

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

OrderStatus

Indicates the current status of an order.

```
FieldIdentifier = 156     ; TSX, TSXV and XATS
OrderStatus     = "F" |   ; Filled
                  "X" |   ; Cancelled
                  "O"     ; Open
```

OrigTradeID

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

```
FieldIdentifier = 506
OrigTradeID     = Trade # "|" Side ("B" = Buy or "S" = Sell
                            or "C" = Combined indicator for both sides)
```

P

PegType

Peg to NBBO. Available on undisplayed orders only.

```
Field Identifier = 597     ; Default = “N”
PegType          = “M”  |  ; Subject to the order’s optional limit price.
                        |  ; Order is priced at the National Best Bid/Offer
                        |  ; midpoint.
                   = “R”  |  ; Primary peg, peg Buy to NBBO Bid, peg Sell to
                        |  ; NBBO Offer (Valid only for XATS).
                   = “N”  |  ; None
```

PortfolioName

Name of the Portfolio assigned by the Client.

```
FieldIdentifier =  218             ; TSX only
PortfolioName   =  1*40AlphaNumeric
```

Price

The limit or type of price for an order.

```
FieldIdentifier = 41
Price           = NumericPrice   |
                  "MKT"          |  ; Market Price ,default
                  "MBF"             ; Must Be Filled, TSX only
Alpha Marketplace: for dark orders, if included, indicates the peg cap price.
```

PrincipalTrade

A transaction where the member as principal sells securities to or buys securities from its particular customer; i.e. a cross between a client and another account type.

```
FieldIdentifier = 71
PrincipalTrade  = "Y" | "N"  ; default is "N"
```

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)
```

PrivateBrokerNumber

True broker number provided in PrivateContent. For anonymous orders, BrokerNumber provided in PublicContent has value of anonymous broker number.

```
FieldIdentifier     = 181
PrivateBrokerNumber = BrokerNumber
```

PrivateBusinessAction

The action to take for a BusinessContent section.

```
FieldIdentifier        = 162 ; no default
PrivateBusinessAction  = "Bought"     |
                         "Buy"        |
                         "Cancelled"  |
                         "Cross"      |
                         "Sell"       |
                         "Short"      |
                         "ShortCross" |
                         "Sold"       |
                         "SoldShort"
```

Note: The PrivateBusinessAction is used in the PrivateContent of messages when the full nature of the business action cannot be disclosed, e.g. Short Sale. In those cases, PrivateBusinessAction supersedes BusinessAction.

PrivateConfirmationType

The type of confirmation for a report.

```
FieldIdentifier = 163
PrivateConfirmationType = "Accepted"             |
                          "AssignTimePriority"   |
                          "Booked"               |
                          "Cancelled"            |
                          "Cross"                |
                          "Frozen"               |
                          "Killed"               |
                          "Pending"              | ; TSX, TSXV and XATS
                          "PriceAssigned"        |
                          "ShortCross"           |
                          "Triggered"              ; TSX, TSXV and XATS
```

Note: The PrivateConfirmationType is used in the PrivateContent of messages when the full nature of the confirmation cannot be disclosed. In those cases, PrivateConfirmationType supersedes ConfirmationType.

PrivateContent

The private business fields contained in a message.

```
PrivateContent  = 1* Field
PrivateContent is one or more Fields comprising the private part of a message.
```

PrivateKeyIdentifier

The business fields for a message.

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

Note: The PrivateKeyIdentifier identifies the entity that has the necessary key codes to decrypt the corresponding PrivateContent in a message.

PrivateNewOrderNumber

True new order number of CFOd order provide in PrivateContent. For anonymous orders, NewOrderNumber provided in PublicContent has value of anonymous new order number. (i.e. OrderNumber associated with anonymous broker number)

```
FieldIdentifier        =  227
PrivateNewOrderNumber  =  OrderNumber
```

PrivateOrderNumber

True order number provided in PrivateContent. For anonymous orders, OrderNumber provided in PublicContent has value of order number associated with anonymous broker number.

```
FieldIdentifier     =  214
PrivateOrderNumber  =  OrderNumber
```

PrivateOrigPrice

The original price type of an order when entered into the trading system with one of the following values:

```
FieldIdentifier    = 507     ; TSX only
PrivateOrigPrice   = "MBF"   ; Must Be Filled, only
```

ProductType

The product type for a symbol.

```
FieldIdentifier = 105
ProductType     = "CDNX-Equity"      | ; TSXV, TMXS and XATS
                  "Equity"           | ; TSX, TMXS and XATS , default
                  "CDNX-Debenture"   | ; TSXV, TMXS and XATS
                  "Debenture”        | ; TSX, TMXS and XATS
```

ProgramTrade

The order is part of a specialized basket trade in securities comprising the S&P/TSX 60 Index to offset an options or futures position.

```
FieldIdentifier = 172       ; TSX, TMXS and XATS
ProgramTrade    = "Y" | "N" ; default is "N"
```

PublicContent

The public business fields contained in a message.

```
PublicContent = BusinessContentChar 1*BusinessContentField
```

PublicPrice

The public price of an order (specifically different than the 'private' price for some pre-open orders.

```
FieldIdentifier = 196
PublicPrice     = Price
```

R

ReasonCode

Additional information for an Order event. This field applies to TSX, TSXV, TMXS and XATS, but code "01" applies to XATS only.

```
FieldIdentifier   = 616
ReasonCode        = "01"       |  ; Killed to avoid freeze, XATS only
                  = "02"-"09"  |  ; Reserved, for internal TMX use only, ignore
                                    if present.
```

ReferenceVolume

The existing volume of the order that is to be CFOd.

```
FieldIdentifier = 43
ReferenceVolume = Volume
```

RegulationId

Identification marker for UMIR specific designations to orders and Trades

```
FieldIdentifier = 325          ; no default
RegulationId    = "IA"         ; InsiderAccount
                  "NA"         ; Not Applicable
                  "SS"         ; Significant Shareholder
```

RemainingBuyParticipationVolume

The remaining buy participation volume for a symbol.

```
FieldIdentifier                = 594     ; TSX only
RemainingBuyParticpationVolume = Volume  ; No default
```

RemainingSellParticipationVolume

The remaining sell participation volume for a symbol.

```
FieldIdentifier                = 595     ; TSX only
RemainingSellParticpationVolume = Volume  ; No default
```

RemainingVolume

Unfilled volume of order.

```
FieldIdentifier = 72
RemainingVolume = Volume
```

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
```

RTAutofill

A marker to indicate a system-generated autofill against the responsible Equities Specialist's account for the TSX or an odd lot trader for TSXV.

```
RTAutofill      = "A" |  ; oddlot
                  "G" |  ; guaranteed fill, TSX only
                  "P"    ; participation, TSX only
```

S

SeekDarkLiquidity

Indicates how the order will trade with dark liquidity.

```
FieldIdentifier   = 613     ; XATS only, no default
SeekDarkLiquidity = "1"  |  ; Trade with dark and lit volume
                  = "2"  |  ; Trade with price improvement dark orders
                  = "3"  |  ; Trade with dark orders only
```

SelfTrade

Indicates that the trade is suppressed from the public feed.

```
FieldIdentifier   = 614     ; XATS only
SelfTrade         = "Y"  |  ; Yes
                  = "N"  |  ; No (default)
```

SellAccountId

The seller's trading account identification.

```
FieldIdentifier = 45
SellAccountId   = AccountId
```

SellAccountType

The seller's type of trading account.

```
FieldIdentifier = 46
SellAccountType = AccountType
```

SellATS-FeeCode

Marker to indicate the sell side of the billing tier code associated with the Alternative Trading System.

```
FieldIdentifier = 396              ; TSX and TMXS
SellATS-FeeCode = 1*3AlphaNumeric   ; no default
```

SellCustomerType

Identifies the cross sell side customer account type.

```
FieldIdentifier  = 510
SellCustomerType = 3Digit      ; 000 to 999, (no default)
```

SellJitney

Marker to indicate that the buy side of a cross is being executed on behalf of another broker.

```
FieldIdentifier = 170
SellJitney      = Jitney
```

SellOrderNumber

The trading system assigned number for the seller's order.

```
FieldIdentifier = 48
SellOrderNumber = OrderNumber
```

SellParticipation

To indicate if the responsible Equities Specialist's participation on the sell side is active.

```
FieldIdentifier   = 166           ; TSX only
SellParticipation = "On" | "Off"  ; no default
```

SellParticipationVolume

To assign the maximum sell participation volume for a symbol.

```
FieldIdentifier         = 593    ; TSX only
SellParticipationVolume = Volume ; no default
```

SellRegulationId

Identification marker for UMIR specific designations to orders and trades.

```
FieldIdentifier = 399
RegulationId    = RegulationId
```

SequenceNumber

The sequence number of the message.

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

SessionType

The function of the session.

```
FieldIdentifier = 149
SessionType     = "EquityFeed"         |
                  "Recovery"
```

SettlementTerms

The terms for settlement of the order.

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

ShortMarkingExempt

Marker for “Short-Marking Exempt” order designation. Required if applicable for “Short-Marking Exempt”.

```
FieldIdentifier = 609     ; no default
ShortMarkingExempt = 0  | ; SME
                     1  | ; Buy Cross SME
                     2  | ; Sell Cross SME
                     3    ; Both Buy and Sell Cross SME
```

SOROrderID1

(SOR) Smart Order Router Order Identifier

```
Field Identifier = 511
SOROrderID1      = 1*15, Alphanumeric  ; no default
```

SOROrderID2

(SOR) Smart Order Router Order Identifier

```
Field Identifier = 512
SOROrderID2      = 1*15,  Alphanumeric  ; no default
```

SourceAddress

The source STAMP address.

```
FieldIdentifier = 54
SourceAddress = DirectedAddress
```

SpecialistName

The stock specialist's or odd lot trader's full name.

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

SpecialistPhoneNumber

Phone number for the Specialist trader or odd lot trader on the stock.

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

SpreadGoal

A unique price range assigned to a stock for purposes of RT spread goal maintenance.

```
FieldIdentifier = 189      ; TSX only
SpreadGoal      = Price
```

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
```

StockIndex

An indicator of membership in a stock index.

```
FieldIdentifier = 176
StockIndex = 1*10AlphaNumeric ; TSX only, default is none
StockIndex = TSE35 |    ; Stock is in TSE35, TSE100, TSE300
           = TSE100 |   ; Stock is in TSE100, TSE300
           = TSE300     ; Stock is in TSE200, TSE300
```

StockState

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

```
FieldIdentifier = 161
StockState      = "Authorized"                          |
                  "AuthorizedDelayed"                   | ; TSX and TSXV
                  "AuthorizedFrozen"                    |
                  "AuthorizedHalted"                    |
                  "AuthorizedPriceMovementDelayed"      | ; TSX and TSXV
                  "AuthorizedPriceMovementFrozen"       | ; TSX and TSXV
                  "Inhibited"                           |
                  "InhibitedDelayed"                    | ; TSX and TSXV
                  "InhibitedFrozen"                     |
                  "InhibitedHalted"                     |
                  "InhibitedPriceMovementDelayed"       | ; TSX and TSXV
                  "InhibitedPriceMovementFrozen"          ; TSX and TSXV
```

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

Time

Time format.

```
Time = 8Digit  ; HHMMSShh (hour, minute, second, hundredths of a second)
```

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: 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. It varies, depending on the type of message:

- Symbol Status message: Total number of symbols per stock group
- Order Book message (TSX and TSXV): Total number of open orders per stock group
- Order Status message (TSX and TSXV): Total number of removed 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 ;TSX and TSXV
TotalNumOpenOrders = 1*8Digit  ; 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
```

TotalVolume

The total volume for undisclosed orders.

```
FieldIdentifier = 226
TotalVolume     = Volume
```

TradeCorrection

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

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

TradeDate

Date that the trade occurred.

```
FieldIdentifier = 109
TradeDate       = Date    ; default is current date
```

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  =  ; YYYYMMDDHHMMSShh (year, month, day, hour
                   ; minute, second, hundredths of a second)
```

TradeThroughExempt

Indicates when a Match is outside the board lot quote.

```
FieldIdentifier   = 392          ; TSX and TMXS
TradeThroughExempt = “Y” | “N”    ; default “Y”
```

TradingTierId

A Trading Engine identifier.

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

TradingSysTimeStamp

The time at which the BusinessAction occurred.

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

U

UndisclTradedVol

The portion of traded volume attributed to the undisclosed volume of an Iceberg order.

```
FieldIdentifier         = 508
UndisclTradedVol        = 1*9Digit   ; (no default)
```

Undisplayed

Indicates that the order is completely undisplayed.

```
FieldIdentifier   = 604             ; Default = "N"
Undisplayed       = "Y" | "N"
```

UserId

The trading system's user id for a trader.

```
FieldIdentifier = 62
UserId          = 1*8AlphaNumeric  ; no default
```

UserOrderId

A unique identifier assigned by the Member Firm to an order.

```
FieldIdentifier = 81
UserOrderId     = 1*16AlphaNumeric  ; no default
```

Note: A STAMP Server will not necessarily verify the uniqueness of `UserOrderId`.

V

VersionNumber

The version number of the TSX/TSXV Broadcast Feed specification used.

```
FieldIdentifier = 65
VersionNumber   = "Version 6.41"
```

Volume

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

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

W

WashTrade

A trade that has occurred between proprietary accounts of the same participating organization.

```
FieldIdentifier = 211
WashTrade       = "Y" | "N"  ; default is "N"
```

Chapter 8 Field IDs by Numerical Order

1	AccountId
2	AccountType
3	ControlClass
4	RESERVED
5	BusinessAction
6	BusinessClass
7	BuyAccountId
8	BuyAccountType
9	RESERVED
10	BuyOrderNumber
11	CFOdOrderNumber
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	ConfirmationType
17	DestAddress
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED
22	RESERVED
23	HeartBeatPeriod
24	RESERVED
25	Jitney
26	RESERVED
27	LowSequenceNumber
28	RESERVED
29	RESERVED
30	MGF-Candidate
31	Discontinued
32	NewOrderNumber
33	RESERVED
34	RESERVED
35	OnStopPrice
36	RESERVED
37	OppositeBroker
38	RESERVED
39	OrderDuration
40	OrderNumber
41	Price
42	BasketTrade
43	ReferenceVolume
44	RESERVED
45	SellAccountId
46	SellAccountType
47	RESERVED
48	SellOrderNumber
49	MGF-Volume
50	SequenceNumber
51	RESERVED
52	RESERVED
53	SettlementTerms
54	SourceAddress
55	Symbol
56	TimeStamp
57	TradingSysTimeStamp
58	Currency
59	RESERVED
60	RESERVED
61	RESERVED
62	UserId
63	RESERVED
64	Volume
65	VersionNumber
66	DISCONTINUED
67	RESERVED
68	RESERVED
69	RESERVED
70	BrokerNumber
71	PrincipalTrade
72	RemainingVolume
73	RESERVED
74	Discontinued
75	ActionSource
76	ExtendedHours
77	RESERVED
78	RESERVED

79	RESERVED
80	StockHaltDate
81	UserOrderId
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	OrderAction
100	RESERVED
101	RESERVED
102	RESERVED
103	RESERVED
104	RESERVED
105	ProductType
106	MessageId
107	RESERVED
108	RESERVED
109	TradeDate
110	AcceptAnonymous
111	NumberOfMessages
112	TotalNumMessages
113	LastMessage
114	LastSale
115	BoardLot
116	RESERVED
117	RESERVED
118	RESERVED
119	FaceValue
120	OpeningTime
121	RESERVED
122	RESERVED
123	RESERVED
124	RESERVED
125	RESERVED
126	RESERVED
127	RESERVED
128	RESERVED
129	Anonymous
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	SessionType
150	DisplayVolume
151	RESERVED
152	MarketType
153	RESERVED
154	RESERVED
155	RESERVED
156	OrderStatus
157	RESERVED
158	RESERVED
159	MarketState
160	MessageText
161	StockState
162	PrivateBusinessAction

163	PrivateConfirmationType
164	DESPrivateContent
165	PrivateKeyIdentifier
166	SellParticipation
167	BuyParticipation
168	NonResident
169	BuyJitney
170	SellJitney
171	CUSIP
172	ProgramTrade
173	Comment
174	RESERVED
175	RESERVED
176	StockIndex
177	SymbolFullName
178	PriorityTimeStamp
179	BidPrice
180	AskPrice
181	PrivateBrokerNumber
182	RESERVED
183	TradeCorrection
184	RTAutofill
185	DISCONTINUED
186	RESERVED
187	RESERVED
188	RESERVED
189	SpreadGoal
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	WashTrade
212	RESERVED
213	RESERVED
214	PrivateOrderNumber
215	RESERVED
216	RESERVED
217	RESERVED
218	PortfolioName
219	RESERVED
220	TradeNumber
221	DISCONTINUED
222	RESERVED
223	RESERVED
224	RESERVED
225	RESERVED
226	TotalVolume
227	PrivateNewOrderNumber
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	Exchange-UserId
266	RESERVED
267	RESERVED
268	RESERVED
269	RESERVED
270	RESERVED
271	RESERVED
272	RESERVED
273	RESERVED
274	OpeningTrade
275	RESERVED
276	RESERVED
277	RESERVED
278	RESERVED
279	RESERVED
280	RESERVED
281	RESERVED
282	StockGroup
283	Market
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	GatewayId
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	RegulationId
326	BuyRegulationId
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	ATS-TimeStamp
351	ATS-Name
352	ATS-SequenceNumber
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	Exchange-Admin
381	RESERVED
382	RESERVED
383	RESERVED
384	RESERVED
385	RESERVED
386	RESERVED
387	RESERVED
388	RESERVED
389	RESERVED
390	CrossType
391	ATS-FeeCode
392	TradeThroughExempt
393	RESERVED
394	RESERVED
395	BuyATS-FeeCode
396	SellATS-FeeCode
397	RESERVED
398	RESERVED
399	SellRegulationId
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	RESERVED
497	RESERVED
498	RESERVED

499	RESERVED
500	RESERVED
501	RESERVED
502	RESERVED
503	ByPass
504	CustomerType
505	NCIB
506	OrigTradeID
507	PrivateOrigPrice
508	UndisclTradedVol
509	BuyCustomerType
510	SellCustomerType
511	SOROrderID1
512	SOROrderID2
513	RESERVED
514	RESERVED
515	RESERVED
516	RESERVED
517	RESERVED
518	RESERVED
519	RESERVED
520	RESERVED
521	RESERVED
522	RESERVED
523	RESERVED
524	RESERVED
525	RESERVED
526	RESERVED
527	RESERVED
528	RESERVED
529	RESERVED
530	RESERVED
531	RESERVED
532	RESERVED
533	RESERVED
534	RESERVED
535	RESERVED
536	RESERVED
537	RESERVED
538	RESERVED
539	RESERVED
540	RESERVED
541	RESERVED
542	RESERVED
543	RESERVED
544	RESERVED
545	RESERVED
546	RESERVED
547	RESERVED
548	RESERVED
549	RESERVED
550	RESERVED
551	RESERVED
552	RESERVED
553	RESERVED
554	ListingMkt
555	RESERVED
556	RESERVED
557	RESERVED
558	RESERVED
559	RESERVED
560	RESERVED
561	RESERVED
562	RESERVED
563	RESERVED
564	RESERVED
565	RESERVED
566	RESERVED
567	RESERVED
568	RESERVED
569	RESERVED
570	RESERVED
571	RESERVED
572	RESERVED
573	RESERVED
574	RESERVED
575	RESERVED
576	RESERVED
577	RESERVED
578	RESERVED
579	RESERVED
580	DELETED
581	TotalNumOpenOrders
582	TotalNumStockGroups

583	TotalNumSymbols
584	TradingTierId
585	RESERVED
586	NoTradeFeat
587	NoTradeKey
588	NoTradeOrderNum
589	NoTradePrice
590	NoTradeVol
591	ExecInst
592	BuyParticipationVolume
593	RemainingBuyParticipationVolume
594	RemainingSellParticipationVolume
595	SellParticipationVolume
596	HandInst
597	PegType
598	MinQty
599	RESERVED
600	RESERVED
601	RESERVED
602	RESERVED
603	RESERVED
604	Undisplayed
605	AcceptUndisplayed
606	IcebergRefresh
607	RESERVED
608	CancelledReason
609	ShortMarkingExempt
610	RESERVED
611	RESERVED
612	COMP
613	SeekDarkLiquidity
614	SelfTrade
615	MatchingPriority
616	ReasonCode
617	IsDark
618	BuyComp
619	SellComp
620	BuySeekDarkLiquidity
621	SellSeekDarkLiquidity
622	AcceptSDL

Chapter 9 Field Names by Alphabetical Order

110	AcceptAnonymous
622	AcceptSDL
605	AcceptUndisplayed
1	AccountId
2	AccountType
75	ActionSource
129	Anonymous
180	AskPrice
391	ATS-FeeCode
351	ATS-Name
352	ATS-SequenceNumber
350	ATS-TimeStamp
42	BasketTrade
179	BidPrice
490	BlindOffsetAccepted
115	BoardLot
70	BrokerNumber
317	BulletinIndicator
5	BusinessAction
6	BusinessClass
7	BuyAccountId
8	BuyAccountType
395	BuyATS-FeeCode
618	BuyComp
509	BuyCustomerType
169	BuyJitney
10	BuyOrderNumber
167	BuyParticipation
592	BuyParticipationVolume
326	BuyRegulationId
620	BuySeekDarkLiquidity
503	ByPass
491	CalculatedClosingPrice
191	CalculatedOpeningPrice
608	CancelledReason
11	CFOdOrderNumber
173	Comment
612	COMP
16	ConfirmationType
3	ControlClass
390	CrossType
58	Currency
171	CUSIP
504	CustomerType
164	DESPrivateContent
17	DestAddress
150	DisplayVolume
247	ExchangeId
380	Exchange-Admin
265	Exchange-UserId
591	ExecInst
76	ExtendedHours
119	FaceValue
311	GatewayId
596	HandInst
23	HeartBeatPeriod
606	IcebergRefresh
492	ImbalanceSide
493	ImbalanceVolume
617	IsDark
25	Jitney
113	LastMessage
114	LastSale
554	ListingMkt
27	LowSequenceNumber
283	Market
197	MarketSide
159	MarketState
152	MarketType
615	MatchingPriority
194	MBX-PartNumber
195	MBX-TotalParts
106	MessageId
160	MessageText
30	MGF-Candidate
284	MGF-Setting
49	MGF-Volume
598	MinQty

494	Moc
495	MocVWAP
32	NewOrderNumber
505	NCIB
168	NonResident
586	NoTradeFeat
587	NoTradeKey
588	NoTradeOrderNum
589	NoTradePrice
590	NoTradeVol
111	NumberOfMessages
35	OnStopPrice
120	OpeningTime
274	OpeningTrade
37	OppositeBroker
99	OrderAction
39	OrderDuration
192	OrderKey
40	OrderNumber
156	OrderStatus
506	OrigTradeID
597	PegType
218	PortfolioName
41	Price
71	PrincipalTrade
178	PriorityTimeStamp
181	PrivateBrokerNumber
162	PrivateBusinessAction
182	RESERVER
163	PrivateConfirmationType
165	PrivateKeyIdentifier
227	PrivateNewOrderNumber
214	PrivateOrderNumber
507	PrivateOrigPrice
105	ProductType
172	ProgramTrade
196	PublicPrice
616	ReasonCode
43	ReferenceVolume
325	RegulationId
593	RemainingBuyParticipationVolume
594	RemainingSellParticipationVolume
72	RemainingVolume
97	Retrans
147	RetransId
184	RTAutofill
613	SeekDarkLiquidity
614	SelfTrade
45	SellAccountId
46	SellAccountType
396	SellATS-FeeCode
619	SellComp
510	SellCustomerType
170	SellJitney
48	SellOrderNumber
166	SellParticipation
595	SellParticipationVolume
399	SellRegulationId
621	SellSeekDarkLiquidity
50	SequenceNumber
149	SessionType
53	SettlementTerms
609	ShortMarkingExempt
511	SOROrderID1
512	SOROrderID2
54	SourceAddress
199	SpecialistName
312	SpecialistPhoneNumber
189	SpreadGoal
282	StockGroup
80	StockHaltDate
176	StockIndex
161	StockState
55	Symbol
177	SymbolFullName
56	TimeStamp
112	TotalNumMessages
581	TotalNumOpenOrders
582	TotalNumStockGroups
583	TotalNumSymbols
226	TotalVolume
183	TradeCorrection

109	TradeDate
220	TradeNumber
392	TradeThroughExempt
264	TradeTimeStamp
57	TradingSysTimeStamp
584	TradingTierId
508	UndisclTradedVol
604	Undisplayed
62	UserId
81	UserOrderId
65	VersionNumber
64	Volume
211	WashTrade

Appendix A Version History

Version	Date	Changes
6.42	2014/01/10	See Release Notes
6.41	2013/03/08	See Release Notes
6.4	2012/11/30	See Release Notes
6.3	2012/04/24	See Release Notes
6.2	2012/02/29	See Release Notes
6.1	2011/07/27	See Release Notes
6.0.2	2011/04/15	See Release Notes
6.0.1	2011/03/28	Defined position 7 of Exchange-Admin as “Execution State” with a marker value of ‘A - Delayed Active’ Added TMXS to CDNX-Equities and CDNX-Debentures values of ProductType tag.
6.0	2011/02/01	See Release Notes
5.6	2010/12/17	See Release Notes
5.5	2010/08/23	See Release Notes
5.4.1	2010/07/21	See Release Notes
5.4	2010/06/04	See Release Notes
5.3.1	2010/04/19	Added [ActionSource] (tag 75), [GatewayId] (tag 311), [Market] (tag 283), and [PrivateKeyIdentifier] (tag 165) to PrivateContent of Trade Report. Removed all “TSXV does not support…” notes.
5.3	2010/03/12	Reformatted. For content changes, see Release Notes.
5.2.2	2009/12/15	See Release Notes
5.2.1	2009/11/19	See Release Notes
5.2	2009/09/30	See Release Notes
5.1.1	2009/09/11	See Release Notes
5.1	2009/07/23	See Release Notes
5.0.5	2009/07/21	See Release Notes
5.0.4	2009/04/09	See Release Notes
5.0.3	2008/12/28	Internal version
5.0.2	2008/10/20	Internal version

Version	Date	Changes
5.0.1	2008/10/02	Expand OrderNumber to 1*18AlphaNumeric Change VersionNumber to 5.0.1 In Field IDs by Numerical Order, correct value of 119 to FaceValue and correct value of 139 to RESERVED In Field Values by Alphabetical Order, correct value of FaceValue to 119
5.0	2008/09/12	Draft Maintenance Version for use by Internal TSX Only until reviewed, approved and published. Please see TSX & TSXV Version 5.0 Release Notes for changes.
4.0.1	2006/07/04	Updates to Disclaimer
4.0	2003/01/26	Release version. Please see CBF & TBF Version 4.0 Release Notes for changes.
3.0.1	2001/11/09	Reformatted and copy edited
3.0	2001/02/15	Release version. Please see TBF Version 3.0 Release Notes for changes.
2.3	1998/01/16	Maintenance version
2.2	1997/07/22	Maintenance version
2.1	1997/03/18	Release version
2.1d6	1997/03/14	Minor cleanup Draft revision advanced to d6 to become in sync with other STAMP spec.
2.1d3	1997/03/11	Additional cleanup changes from internal review of document
2.1d2	1997/01/29	Additional cleanup changes from internal review of document
2.1d1	1997/01/22	Maintenance version including TOREX requests
2.1	1996/09/06	Maintenance version.
2.0	1996/07/29	Release version. Specification name changed from TSE Broadcast Feed Specification to TOREX Broadcast Feed Specification (TSE Only).
1.0	1996/05/01	Release version.

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