



## Multicast Market Data Feed

## Specifications

Version 1.3

Revision Date: March 2018

TriAct Canada Marketplace LP  
130 King Street West, Suite 1050  
Toronto, Ontario M5X 1B1  
416-861-1010  
[www.triactcanada.com](http://www.triactcanada.com)

Copyright 2005-2017 TriAct Canada Marketplace LP (TriAct), Member of the Investment Industry Regulatory Organization of Canada Inc. ("IIROC") & Canadian Investor Protection Fund ("CIPF"), is registered as an Investment Dealer in the Provinces of Ontario and Alberta, and exempt from registration as a dealer in the Provinces of BC, SK, MN, QC, NS, NB and NFLD & Labrador. This document has been prepared for informational purposes only, and does not constitute a solicitation to buy or sell securities. TriAct does not warrant the completeness or accuracy of the information provided herein.

REVISION HISTORY

Date	Version	Author	Description
2017/04/27	1.0	Zeeshan Qazi	Initial version.
2017/06/14	1.1	Vince Poil	Language and format changes
2017/11/16	1.2	Vince Poil	Spec change - Added Primary Exchange Code to the feed and updated trade example
2018/03/07	1.3	Vince Poil	Added network Specs and wording and corrected typo in trade message

CONTENTS

1 INTRODUCTION .....4

2 MATCHNow’s MARKET DATA BINARY ITCH MULTICAST .....5

3 CONFIGURATION INFORMATION .....6

4 PROTOCOL PRIMITIVES .....7

5 APPLICATION MESSAGES .....8

6 RETRANSMISSION SESSION AND PROTOCOL .....9

7 SAMPLE PACKETS .....12

## 1 INTRODUCTION

This document provides the specifications for the Binary ITCH Protocol over Multicast IP used for dissemination of the public trade data for MATCHNow along with the protocol used for recovery of missed data over TCP/IP.

This protocol is loosely based on Multicast ITCH Protocols used by other Canadian and international marketplaces. This document describes the protocol in detail and does not require a supplement for interpretation of this protocol.

The data feed published in this protocol may be published on distinct Multicast Groups from distinct sources. Though the data from two sources may be identical, individual messages may not be ordered in the same way. If two multicast groups are being published from the same source, then the data on them would be identically ordered.

The primary Feed consists of two feed sauces. The secondary sauce would only start publishing after the primary sauce is down for 15 seconds. From the point of view of the subscriber, the only noticeable discrepancy will be the 15 second gap in the data stream.

## 2 MATCHNow's MARKET DATA BINARY ITCH MULTICAST

Since MATCHNow only has a dark order book, the Binary Itch Multicast feed will only contain trades and busts. The multicast feed is session free and requires networking to allow PIM forwarding from the client network to the MATCHNow network to be able to subscribe to the feed. The recovery mechanism requires a TCP session to be established and a sequence range to be queried. The recovery mechanism is meant to be used to recover messages that were missed on the multicast feed. Each recovery range requires the client to establish a TCP session and ask for a range separately. A client may have multiple such sessions open simultaneously.

Upon request, our MATCHNow Support staff will provide you with access to the test Multicast Feed and recovery server. An appointment is required for certification purposes.

While using the test feed, the client will receive approximately the same behavior as the production feed, however, the trade data on the test feed will be from the test environment, as opposed to the production environment.

### **Communications**

Clients may access MATCHNow Binary Itch Multicast either through leased lines or a dedicated network providers.

### **Full Certification**

A Multicast Feed is passively receivable so does not require any certification other than basic verification that the data is received and translated correctly. The retransmission feed does require certification to ensure that interfacing with our feed provider is correctly implemented on the client side.

## 3 CONFIGURATION INFORMATION

### 3.1 Client Configuration

The client will be provided with a bundle consisting of following information:

Value	Description
Primary Node Primary Multicast Address	The multicast address that the client must join to receive the data from the primary node.
Primary Node Primary Multicast Port	The port on which the client must listen for the incoming feed.
Primary Node Retransmission Address	The address on which to establish a retransmission session over TCP.
Primary Node Retransmission Port	The port on which the client must connect over TCP to the Retransmission Address above.
DR Node Primary Multicast Address	The multicast address published from the DR site, This feed is not identical to the feed published from the primary site in terms of ordering of messages though identical data is published from this site as well.
DR Node Primary Multicast Port	The port on which the client must listen for the DR Multicast Feed.
DR Node Retransmission Address	The address on which to establish a retransmission session over TCP to recover missed messages from the DR feed.
DR Node Retransmission Port	The port on which the client must connect over TCP to the Retransmission address for the DR Node.

### 3.2 MATCHNow Configuration

MATCHNow multicast feed and retransmission servers add a source identifier to the packet headers on the multicast feed and retransmission response messages respectively. These source identifiers aid in determining whether two sources have identical data or not. The source identifier is a 4 character string, with first three characters matching, if and only if, two sources have the same data, and the fourth character represents an internal node number.

For Instance:

- MR11 - Markham Feed Source 1, Primary Node [Feed A]
- TO11 - Toronto Feed Source 1, Primary Node [Feed B]

### 3.3 TCM Feed IP/Port Specification

TCM Feed Name	Location	Feed Src IP	Feed MC IP	Feed UDP Port	RP	Feed Recovery IP	Recovery tcp port
UAT Feed A	DC2 -100 Wellington Ave. W.(TOR); 151 Front Str W.(POP)	45.40.27.70/32 45.40.27.71/32	224.0.159.195	13305	45.40.27.2	45.40.27.11	13405
Prod Feed A	DC1 - 3500 Steeles Ave. E. (MRK); 151 Front Str W.(POP)	45.40.28.5/32 45.40.28.6/32	224.0.159.210	13317	45.40.28.120	45.40.28.115	13417
Prod Feed B	DC2 -100 Wellington Ave. W.(TOR); 151 Front Str W.(POP)	45.40.28.133/32 45.40.28.134/32	224.0.159.225	13318	45.40.28.248	45.40.28.243	13429

## 4 PROTOCOL PRIMITIVES

MATCHNow Binary ITCH multicast protocol is a binary protocol, this means that the data on the wire is in network byte ordering and contains binary representation of numbers. Fields in this protocol are fixed width. Each UDP packet may contain one or more market data messages. Heartbeats are UDP packets that contain zero market data messages and just the next sequence number expected from the source.

### 4.1 Data Types

Type	Size in bytes	Description
NetUInt16	2	16-bit unsigned big-endian integer
NetUInt32	4	32-bit unsigned big-endian integer
NetUInt64	8	64-bit unsigned big-endian integer
Price	4	32-bit unsigned big-endian integer scaled by a factor of 10,000  Price value is arrived by dividing this number by 10,000
AlphaNumeric [ N ]	N	Left-justified Space Padded Alphanumeric string.

### 4.2 Packet Header

Each Multicast Packet consists of a header identifying the number of business messages in the packet and the source of those messages.

Offset	Field Name	Type	Description
0	Sequence	NetUInt32	Sequence number of the next message
4	MessageCount	NetUInt16	Total number of messages in this packet
6	SourceIdentifier	AlphaNumeric[4]	Identifier of the source that published this data

The packet header itself is just 10 bytes. If the MessageCount in the header is zero, then the packet represents heartbeat. Any data in the packet after the messages (indicated by MessageCount) must be ignored, as this data is used internally by MATCHNow. The packet header is only applicable to the **Multicast Feed** and not to the **Retransmission Session**.

### 4.3 Message Header

Each business message in a packet starts with the message header, describing the length of the message. For the purpose of this document this header is included in the business messages. Here it is being provided to you as a reference to understand how messages are framed.

Offset	Field Name	Type	Description
+0	MessageLength	NetUInt16	Length of the Message excluding the length field
+2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
+10	MessageType	AlphaNumeric[1]	Message Type Identifier

The message header is 11 bytes in size. The plus (+) in the offset above means that the position in the packet is after the previous component in the packet itself. This header is used in both the **Multicast Feed** and the **Retransmission Session** to define boundaries of the messages.

## 5 APPLICATION MESSAGES

This section discusses the application-level Binary ITCH messages that describe the trade events that occur on MATCHNow. These messages are identical both on the Multicast Feed as well as on the retransmission feed. The message header is actually incorporated into the messages themselves, though the packet header used in the multicast feed is not, since it is shared for the entire packet. In the future these messages may grow in backwards compatible fashion, which means that new fields may be added and therefore the length of the messages may change to incorporate new fields that are appended to the end of the message.

### 5.1 MATCHNow Market Data Messages

These messages are used to disseminate the public trade information for the trades that occur on MATCHNow.

#### 5.1.1 Trade Message

This message represents a public trade that occurred on MATCHNow.

Offset	Field Name	Type	Description
+0	MessageLength	NetUInt16	Length of the Message excluding the length field = 58
+2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
+10	MessageType	AlphaNumeric[1]	Message Type Identifier = 'T'
+11	Side	AlphaNumeric[1]	Side of the transaction. 'B' (only Buy side sent)
+12	LastShares	NetUInt32	Shares traded in last trade
+16	Stock	AlphaNumeric[10]	Ticker Symbol for the stock
+26	Listing Exchange	AlphaNumeric[4]	MIC code of listing exchange: NEOE (Aequitas) XCNO (CSE) XTSE (TSX) XTSX (TSX Venture)



+30	LastPrice	Price	Price of the last trade
+34	TradeReference	AlphaNumeric[20]	Unique Identifier for the trade
+54	Broker	NetUInt16	Broker Number (or 1 for anonymous)
+56	ContraBroker	NetUInt16	Contra-Broker (or 1 anonymous)
+58	NodeIndex	NetUInt16	Internal Node Identifier as source of this message

The size of this message including the length field is currently 60 bytes. It may grow in future versions, but the portion of the message described in this document will stay the same.

### 5.1.2 Bust Message

This message represents a public trade bust that occurred on MATCHNow.

Offset	Field Name	Type	Description
+0	MessageLength	NetUInt16	Length of the Message excluding the length field = 58
+2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
+10	MessageType	AlphaNumeric[1]	Message Type Identifier = 'B'
+11	Side	AlphaNumeric[1]	Side of the transaction. Always 'B'
+12	LastShares	NetUInt32	Shares traded in last trade
+16	Stock	AlphaNumeric[10]	Ticker Symbol for the stock
+26	Listing Exchange	AlphaNumeric[4]	MIC code of listing exchange: NEOE (Aequitas) XCNQ (CSE) XTSE (TSX) XTSX (TSX Venture)
+30	LastPrice	Price	Price of the last trade
+34	TradeReference	AlphaNumeric[20]	Unique Identifier for the trade that is being cancelled
+54	Broker	NetUInt16	Broker Number (or 1 for anonymous)
+56	ContraBroker	NetUInt16	Contra-Broker (or 1 anonymous)
+58	NodeIndex	NetUInt16	Internal Node Identifier as source of this message

The size of this message is currently 60 bytes. It may grow in future versions, but the portion of the message described in this document will stay the same.

## 6 RETRANSMISSION SESSION AND PROTOCOL

### 6.1 Retransmission Session

Each time there is a retransmission needed by a client, they need to open a TCP/IP connection to the IP and port provided for the retransmission server, and send a retransmission request. The server either responds with a retransmission response followed by requested messages (or a sub-range if the range was beyond known messages) or a reject message indicating an error.

### 6.2 Retransmission Messages

These messages are additional messages that are exchanged as part of the retransmission request mechanism. There are no heartbeats on the retransmission session, and it is only expected to last only for so long as is required to retrieve the missed messages.

### 6.2.1 Retransmission Request Message

This message is sent on the retransmission session by the client to request a range of messages. This is done right after the connection is established with the retransmission host/port.

Offset	Field Name	Type	Description
0	MessageLength	NetUInt16	Length of the Message excluding the length field = 17
2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
10	MessageType	AlphaNumeric[1]	Message Type Identifier = 'R'
11	StartSequence	NetUInt32	Start Sequence Number
15	EndSequence	NetUInt32	End Sequence Number

The size of this message is 19 bytes.

### 6.2.2 Retransmission Response Message

This message is sent on the retransmission session by the server to the client to indicate acceptance of a retransmission request. This is done in response to the retransmission request, and is followed by the application messages that the server agreed to provide in this message. The range originally requested may be adjusted to what is known presently by the server. The client is given the ability to over-request to provide them ability to be able to get additional messages that may be dropped in the time-frame of the retransmission attempt.

Offset	Field Name	Type	Description
0	MessageLength	NetUInt16	Length of the Message excluding the length field = 21
2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
10	MessageType	AlphaNumeric[1]	Message Type Identifier = 'w'
11	StartSequence	NetUInt32	Start Sequence Number
15	EndSequence	NetUInt32	End Sequence Number
19	SourceIdentifier	AlphaNumeric[4]	Source of the messages

The size of this message is 23 bytes.

### 6.2.3 Retransmission Reject Message

This message is sent to reject a retransmission request by the server to the client.

Offset	Field Name	Type	Description
0	MessageLength	NetUInt16	Length of the Message excluding the length field = 137
2	TimeStamp	NetUInt64	Timestamp of the business message. Microseconds since midnight for the day in UTC.
10	MessageType	AlphaNumeric[1]	Message Type Identifier = 'j'

11	RejectText	AlphaNumeric[128]	Reason for rejecting request
----	------------	-------------------	------------------------------

The size of this message is 139 bytes.

## 7 SAMPLE TRADE PACKET

Just to help you understand how the data is disseminated here is a broken-down trade message.

00:00:00:04:00:02:4d:52:4b:31:00:3a:00:00:00:0b:50:cb:4c:c8:54:42:00:00:01:2c:56:52:58:20:20:20:20:20:20:58:54:53:45:00:09:7f:40:32:30:30:33:30:30:30:31:30:37:39:31:38:4d:32:30:30:30:30:35:00:02:00:02:00:00:00:3a:00:00:00:0e:07:a8

Packet Header

- ⇒ Next Sequence = 4
- ⇒ Total Messages = 2
- ⇒ Message Source = MRK1

Message Header

- ⇒ Message Length = 58 (0x36)

Trade Message

- ⇒ TimeStamp = 48600141000 (0x0000000b50cb4cc8) → 13:30:00.141000 UTC
- ⇒ Message Type = T (0x54)
- ⇒ Side = B (0x42)
- ⇒ Size = 300 (0x012c)
- ⇒ Symbol = VRX
- ⇒ Primary Exchange = XTSE
- ⇒ Price = 62.64 (encoded as: 626400, 0x004d3230)
- ⇒ TradeReference = 2003000107918M200005
- ⇒ Broker = 1 (0x0002)
- ⇒ ContraBroker = 1 (0x0002)
- ⇒ Node = 0 (0x0000)

Internal Data → Ignored by client

And here is a packet capture showing one packet on the network containing a pair of trades:

0000	01 00 5e 00 9f c3 44 4c a8 0e ce 99 81 00 00 76	..^...DL .....v
0010	81 00 01 c0 08 00 45 00 00 a6 00 00 40 00 3d 11	.....E. ....@.=.
0020	75 14 2d 28 1b 47 e0 00 9f c3 d4 66 33 f9 00 92	u.-(.G.. ...f3...
0030	96 dc 00 00 00 04 00 02 4d 52 4b 31 00 3a 00 00	..... MRK1:...
0040	00 0e 07 a8 2c 20 54 42 00 00 01 2c 56 52 58 20	...., TB ...,VRX
0050	20 20 20 20 20 20 58 54 53 45 00 03 56 7e 32 30	XT SE..V~20
0060	30 33 30 30 30 31 30 37 39 31 38 4d 32 30 30 30	03000107 918M2000
0070	30 35 00 02 00 02 00 00 00 3a 00 00 00 0e 07 a8	05..... :.....
0080	2c 20 54 42 00 00 00 c8 56 52 58 20 20 20 20 20	, TB.... VRX
0090	20 20 58 54 53 45 00 03 56 7e 32 30 30 33 30 30	XTSE.. V~200300
00a0	30 31 30 37 39 31 38 4d 32 30 30 30 30 36 00 02	0107918M 200006..
00b0	00 02 00 00 dc 32 00 00 00 00 00 00 00	.....2.. ....