DOCUMENT TITLE

Universal Trading Platform Market Data Feed

Generic Customer Specifications

APPLICATION VERSION

STATUS

Version 1.3

23 April 2009

Introduction

Copyright Trademark Statements

Overview

This document details the specifications of the Universal Trading Platform Market Data Feed. This specification is for developers that wish to write applications that interface with the Universal Trading Platform Market Data Service.

Associated documents

The complete set of Universal Trading Platform Market Data Feed (UTP-MD) Specifications include a number of appendices, supplying detailed information about message structure and contents, which should be read in conjunction with this document

History

V 1.0 16 May2008: Initial version
V 1.2 25 September 2008:
- Add feed processing guidelines, clarifications where necessary, remove refresh functionality.

V1.3 23 April 2009
- Add High Availability Retransmission Behaviour (chapter 3.2)

Table of Contents

INTRODUCTION 2
Copyright Trademark Statements *2*
Overview *2*
Associated documents *2*
History *2*
TABLE OF CONTENTS 3

1 INTRODUCTION 4
1.1 OVERVIEW 4
1.2 ACCESS TO DATA 4

2 REAL-TIME MARKET DATA 5

3 DATA RETRANSMISSION 6
3.1 RETRANSMISSION SERVER HEARTBEAT 7
3.2 HIGH AVAILABILITY RETRANSMISSION BEHAVIOUR 7

4 PROCESSING GUIDELINES 8
4.1 GENERAL PROCESSING NOTES 8
4.2 PACKET STRUCTURE 8
4.3 COMPRESSION 9
4.4 SEQUENCE NUMBERS 10
4.5 DETECTING AND RECOVERING MISSED DATA 10
4.5.1 Gap Detection *10*
4.5.2 Line Arbitration *12*
4.5.3 Retransmission Server *12*

5 OPERATIONAL INFORMATION 14
5.1 EXCHANGE SYSTEM FAILURE 14
5.2 CLIENT SYSTEM FAILURE 14
5.3 GAP DETECTION 14
5.4 FEED OPERATION HOURS 14

1 INTRODUCTION

1.1 Overview

The Universal Trading Platform Market Data Feed provides high speed real-time market data for NYSE Euronext markets.

The feed has the following high-level features:

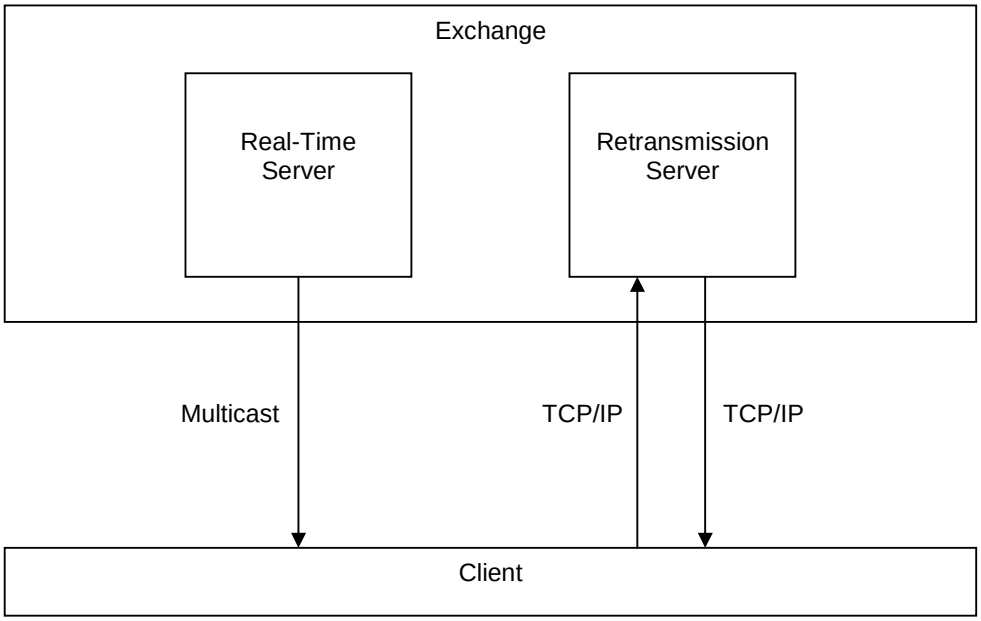
- Multicast technology
- Optional FAST-based compression
- High system availability
- Ultra-low latency
- Reliable network solution
- High level of scalability
- Access to wide range of US and European market data sets

This document provides detailed information about the features of the feed, to support the development of client applications by Members, Independent Software Vendors and Quote Vendors.

Additional documents provide details that are specific to each of the NYSE Euronext market data sets, including formats for each message type.

1.2 Access to Data

Customers connect to multicast addresses for the real-time market data messages, and can also connect to a TCP/IP server for packet retransmissions. Requests for retransmission are performed via TCP/IP.



2 REAL-TIME MARKET DATA

Real-time market data is message-based over the UDP IP protocol with fixed length binary and ASCII fields.

It uses the push-based publishing model. This means that data will be published based on its availability. Once an update is available, it will be published to the appropriate multicast group.

For capacity reasons, market data will be split across a number of multicast groups organised into predefined data sets.

Each multicast group will deliver a set of data for a certain market segment.

The client application will be responsible for issuing Multicast subscriptions to one or more of the Multicast Groups assigned to each product.

The process of subscribing to a Multicast Group ID is also known as ‘joining’ a Multicast Group. Upon session termination, the subscriber’s host system should issue an ‘unjoin’ message. This will terminate delivery of data to that host’s local network. If a client application terminates without issuing an ‘unjoin’ message, the network will eventually issue a ‘timeout’ for the Multicast Group subscription that will automatically terminate delivery of the Multicast packets to the host’s local network.

The ‘join’ and ‘unjoin’ processes are standard functions. No specific instructions are provided here, as they are specific to the user’s operating system and programming language.

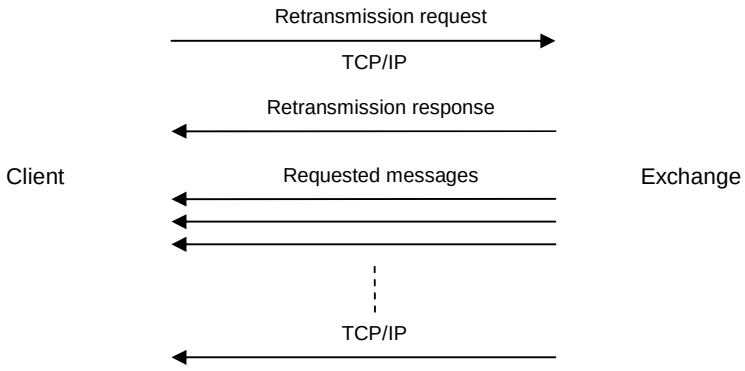
3 DATA RETRANSMISSION

The retransmission functionality is designed to allow the user to recapture a small number of missed packets.

It is not intended that clients use the retransmission functionality to recover data after long outages or on late start up. Accordingly, the number of packets that the user can request is strictly limited.

Equally, the number of retransmission requests permitted per user is limited per day.

The client makes a TCP/IP connection with the retransmission server, and receives the requested messages also via the TCP/IP channel.



The retransmission request will include a username which will be validated by the exchange system.

The request may be rejected for any of the following reasons:

- Invalid username;
- Invalid packet sequence number;
- Total number of packets requested in the current day exceeds the predefined system limit;
- Number of retransmission requests in the current day exceeds the predefined system limit.

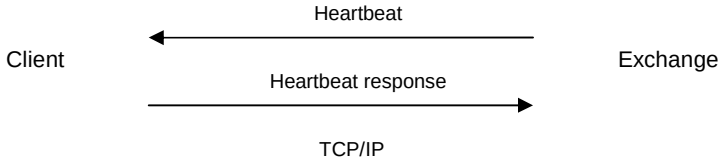
In the case of such a failure, the user will receive an error message to advise of the reason for failure.

3.1 Retransmission Server Heartbeat

Users can choose to either disconnect following each retransmission request, or remain connected to the Retransmission Server.

To determine the health of the user connection on the TCP/IP link, the Retransmission Server will send regular heartbeat messages to the user. The heartbeat frequency is 30 seconds.

The user must respond to the Heartbeat message with a Heartbeat Response message should they wish to remain connected. Such a response has to be received by the server within 5 seconds of a heartbeat message.



3.2 High Availability Retransmission Behaviour

There are two redundant retransmission servers at the exchange. Customers should monitor the connection to the retransmission server to determine if there is an outage, and if there is they have the ability to connect to the 'secondary' retransmission server in the event of a failover. The secondary retransmission server will maintain the same cache of packets as the primary therefore providing redundancy. Customers should monitor the availability of the primary / secondary retransmissions server by the following means:

- For customers remaining connected to the retransmission server throughout the day, a disconnection from the retransmission server should trigger a failover to the secondary retransmissions server.
- Customers may choose to only connect to the retransmissions server if the application requires packets to be serviced. In this instance, customers should failover to the secondary retransmission server if they cannot establish a connection with the primary.

4 PROCESSING GUIDELINES

4.1 General Processing Notes

The following processing notes apply to the messages sent through the feed:

- All fields will be sent for every packet;
- Only field values will appear in the published messages (e.g., no names or 'tags' will appear in the message);
- The field names that appear in the message format documents are for reference purposes only;
- All the fields are contiguous, with reserved fields for alignment issues;
- All field sizes are fixed and constant;
- Binary fields are provided in network byte order (Big Endian format);
- ASCII string fields are left aligned and null padded;
- Segmentation of messages across packets will not be supported. This means a message will never straddle a packet boundary.

4.2 Packet Structure

All packets of data sent on the Universal Trading Platform Market Data Feed will have a common packet header followed by one or more messages (with the exception of some technical messages that do not contain any messages).

The packet header format is the same for all packets, and contains packet length, number of messages within the packet, packet sequence number etc.

The format of each message in the packet depends on message type, but each message will start with message size and message type.

The maximum length of a packet is 1400 bytes.

A packet will only ever contain complete messages. A single message will never straddle multiple packets.

The message size will never exceed the maximum packet length (less the packet header size).

Packet Header	Message 1	Message 2	…	Message n

The packet header provides information including the total packet length, a packet sequence number, the number of messages within the packet and a send timestamp. The format is as follows:

Packet Header				
Field	Offset	Length	Format	Description
PacketLength	0	2	Binary Integer	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Integer	Identifier for the type of data contained in the packet.
PacketSeqNum	4	4	Binary Integer	Packet sequence number, unique for each broadcast stream.

SendTime	8	4	Binary Integer	SourceTime in milliseconds indicating the packet broadcast time. The number represents the number of milliseconds since the previous Sunday 00:00 UTC. Indicates the time the message was sent by the market data feed.
ServiceID	12	2	Binary Integer	Numeric value identifying the broadcast stream.
DeliveryFlag	14	1	Binary Integer	Indicates delivery method. '0' – Real Time message (Uncompressed) '2' – Retransmission message (Uncompressed) '8' – Real Time message (Compressed using FAST) '10' – Retransmission message (Compressed using FAST)
NumberMsgEntries	15	1	Binary Integer	The number of messages that are contained within the packet.

The format of each message within a packet will vary according to message type. However, regardless of the message type, each message will start with a 2-byte message length followed by a 2-byte message type.

Message				
Field	Offset	Length	Format	Description
MsgSize	-	2	Binary Integer	Size of the message in bytes, excluding these two bytes.
MsgType	-	2	Binary Integer	Type of message (refer to appendices for further details).
Additional fields specific to each message type …				

Users should refer to the relevant appendix for information on the format of each message.

4.3 Compression

FAST compression may be enabled for certain services. Users should refer to the relevant appendix for information about compression for each service.

Note that the packet header will never be compressed. The DeliveryFlag field in the packet header indicates whether the messages in the packet are compressed or not.

4.4 Sequence Numbers

All messages conform to the line level sequencing. Each channel has its own packet sequence number. Subscribers can use packet sequence numbers to determine the following:

- Missing (gapped) packets
- Unordered packets
- Duplicate packets

Clients should note that the packet sequence number per channel might restart from one following a failure recovery. A reset packet sequence number message will be sent to clients via the Multicast Groups to inform of such an event.

4.5 Detecting and Recovering Missed Data

UDP can at times be unreliable and may drop packets from both the primary and secondary data feeds.

The UTP Market Data Feed provides 2 different mechanisms for recovering missed data:

- Line arbitration – using dual multicast channels
- Retransmission server – recovery of limited number of packets

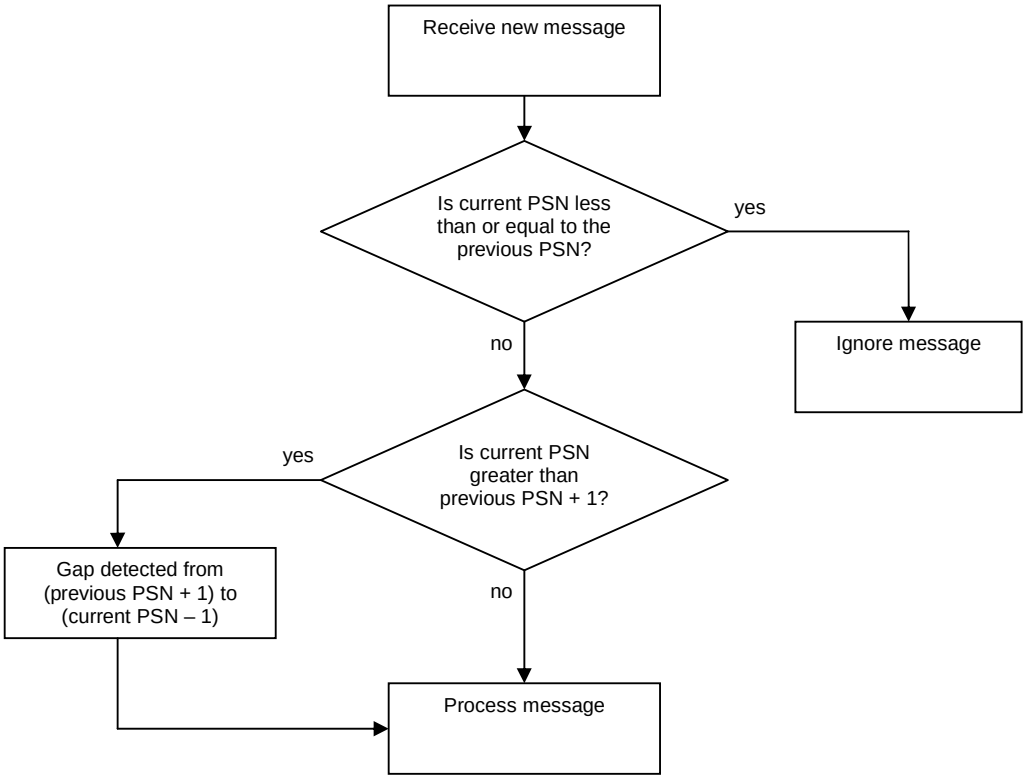
These mechanisms should be used as follows:

Event	Action
Dropped packet(s) on primary multicast channel	Recover from other channel using line arbitrage
Dropped packet(s) on both primary and secondary multicast channels	Recover dropped packet(s) from retransmission server

4.5.1 Gap Detection

Each packet has a Packet Sequence Number (PSN). PSNs start at one (1) and increase monotonically (one by one and without gaps) with each subsequent message. Users should use the PSN to detect gaps in the transmission of messages.

The following diagram illustrates how the PSN should be used to detect gaps in the feed.


Receive new message
Is current PSN less than or equal to the previous PSN?
yes
Ignore message
no
Is current PSN greater than previous PSN + 1?
yes
Gap detected from (previous PSN + 1) to (current PSN – 1)
no
Process message

4.5.2 Line Arbitration

Client applications should check the Packet Sequence Number (PSN) for every packet received.

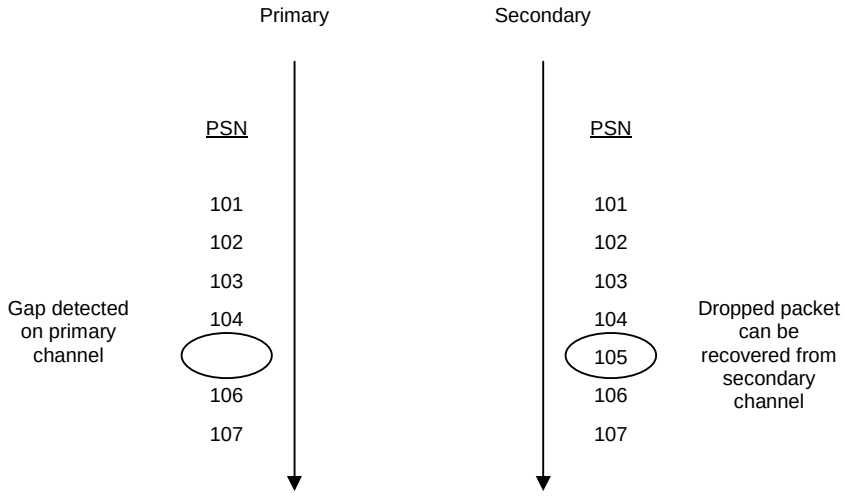
PSNs are unique and increase monotonically for each service.

The primary and secondary channels are identical in terms of:

- Packet contents
- PSNs
- Sequence in which packets are sent

In the event a packet is lost on the primary channel for a multicast group, clients can retrieve the lost packet from the secondary channel.

As a first resort, clients should use the secondary channel to fill gaps on the primary channel, as shown in the following diagram:



4.5.3 Retransmission Server

If a packet is lost from both the primary and secondary channels, clients then make a TCP/IP request to have the packets resent. Packets are resent from the Retransmission Server.

After a client establishes a TCP/IP connection, the Retransmission Server will periodically send heartbeat request messages to the client. Clients must respond to this request with a heartbeat response within a specific timeframe – otherwise, the Retransmission Server will close the connection. This timeframe is currently set to thirty seconds but is subject to change—so clients should make this configurable. (Clients will be informed of changes to the timeframe via customer notice.)

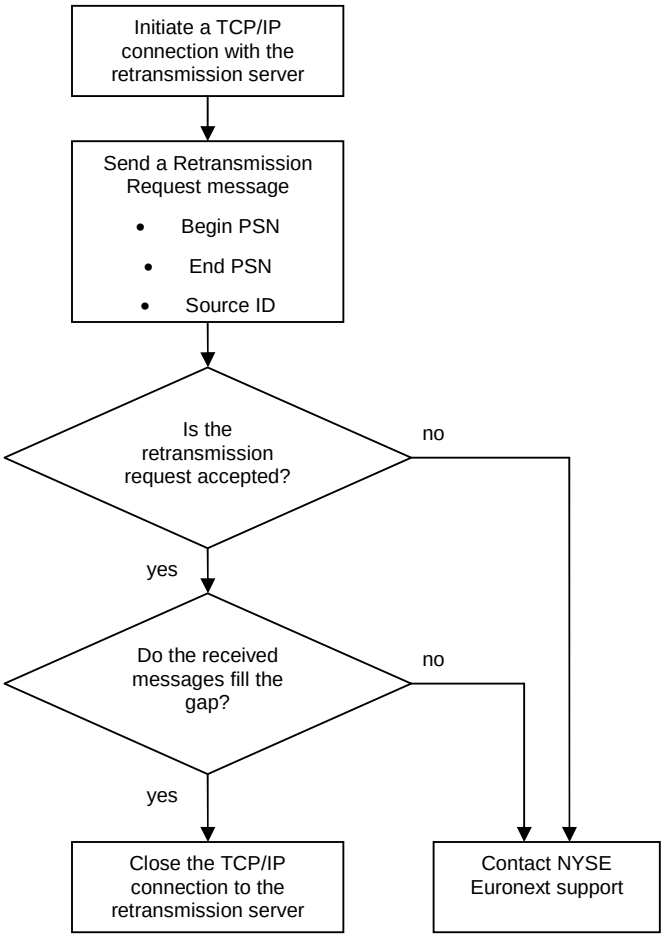
The client makes a TCP/IP connection to the Retransmission Server for both requesting and receiving retransmitted packets.

Retransmission requests should contain a Start PSN, an End PSN and a Source ID. The Source ID identifies the client application, and will be supplied by the exchange. The request will be rejected if an invalid Source ID is supplied. Each Source ID may only be logged in once per port at any given time.

The number of retransmissions allowed per client per day is limited.

The length of each retransmission is limited to a pre-defined number of packets.

The following diagram illustrates the process of requesting dropped packets from the retransmission server:



5 OPERATIONAL INFORMATION

The exchange system is designed to be extremely resilient. However there are measures in place to safeguard against unexpected system failures.

5.1 Exchange System Failure

Under normal operating conditions, the exchange system will send real-time messages to two unique multicast addresses. This provides customers with two redundant data feeds. The client application should be designed to handle the loss of one of the two multicast channels without any interruption to service.

5.2 Client System Failure

Real-time market data will be made available on two different multicast groups. This offers clients the possibility to set up more than one receiving system processing the same data. In the event of a client system failure, the backup client system should continue to process the real-time data sent on the second multicast group.

5.3 Gap Detection

The Universal Trading Platform Market Data Feed provides a unique, sequential packet sequence number for each multicast channel. This will allow recipients to identify ‘gaps’ in the message sequence and, if appropriate, reconcile them ‘locally’ with an alternate channel or request retransmission of the missing/corrupted data packet.

Refer to section 4.5.1 for more details on gap detection.

5.4 Feed Operation Hours

Please refer to supporting documentation for each data set for the operational hours of each service.