

MoldUDP64 Protocol Specification

V 1.00

07/07/2009

Date	Author	Notes
11/2/04	SL	Initial version.
12/4/08	SL	Updated for 64-bit support
2/24/09	HT	Corrected minor error in documentation
7/7/09	SM	Clarified the data length field definition

Addressing

Note: For the current list of IP Addresses for NASDAQ OMX MoldUDP and MoldUDP64 protocol data feeds, please refer to <http://www.nasdaqtrader.com/Trader.aspx?id=FeedIPS>.

Overview

MoldUDP64 is a networking protocol that allows efficient and scaleable transmission of data messages in a “one transmitter to many listeners” scenario. MoldUDP64 is a lightweight protocol layer built on top of UDP that provides a mechanism for listeners to detect and re-request missed packets.

In MoldUDP64, each outbound packet is transmitted only once regardless of the number of listeners. Multiple messages may also be aggregated into a single network packet to reduce network traffic. Optional caching Re-request Servers can be placed nearby remote receivers to reduce latency and bandwidth over WAN links.

This document describes the messages sent between a MoldUDP64 server and its clients. MoldUDP64 transmitters send Downstream packets via UDP multicast to transport the normal data stream sent to the listeners. These packets are also sent via UDP unicast in response to a Request message submitted by a listener. MoldUDP64 clients can send these Request messages to request the retransmission of any desired packets from the data stream.

The MoldUDP64 server will transmit on a well known multicast group for each type of downstream MoldUDP64 datastream on a network. The listeners must subscribe to this multicast group to receive the downstream data. One or more Re-request Servers may also be deployed to service any unicast client requests for retransmission of specific messages. The listeners must be configured with these IP addresses and port combinations to which they can submit the requests.

Assumptions

All number fields in the MoldUDP64 messages specified in this document (i.e. sequence number, message counts and message lengths etc) are binary numbers formatted in Big Endian mode (i.e. most significant byte first). Note: This need not apply to the data contained the Message Data fields of the Message Blocks.

Terms

Message

A message is an atomic piece of information carried by the MoldUDP64 protocol.

MoldUDP64 can theoretically handle individual messages from zero bytes up to 64KB in length although individual messages should be kept small enough so that the UDP underlying network protocol can efficiently carry the resulting MoldUDP64 packets.

The contents of a MoldUDP64 message are defined by the higher level application.

Session

A Session is a sequence of one or more messages.

While a single session can last indefinitely, typically the application will define a session to logically group messages together based on time delimitation.

Once a session is terminated, no more messages can be sent on that session. Depending on the design of the MoldUDP64 system and the application, receivers may still be able to re-request messages from a terminated session.

A session is considered active if it has started but not yet been terminated.

Downstream Packet

A MoldUDP64 transmitter sends “downstream” packets that are received by MoldUDP64 listeners. A MoldUDP64 packet may contain a payload of 0 or more data stream messages.

Each MoldUDP64 packet consists of a Downstream Packet Header and of a series of Message Blocks. The Message Blocks carry the actual data of the stream.

Header

Downstream Packet Header

Field Name	Offset	Len	Value	Notes
Session	0	10	ANUM	Indicates the session to which this packet belongs.
Sequence Number	10	8	NUM	The sequence number of the first message in the packet.
Message Count	18	2	NUM	The count of messages contained in this packet.

Sequence Number

The Sequence Number field of the packet Header indicates the sequence number of the first message in the packet. If there is more than one message contained in a packet, any messages following the first message are implicitly numbered sequentially.

Message Count

The number of Message Blocks contained in a MoldUDP64 packet is specified by the Message Count field of the Packet Header. The maximum payload size of a Downstream Packet is determined by the sender. Note that a Message Count of zero denotes a heartbeat and that a Message Count of 0xFFFF(hex, or 65535 in decimal) denotes end of session.

Message Block

The first field of a Message Block is the two byte Message Length. The remainder of the Message Block is the variable length Message Data field. The first Message Block field will always start immediately following the Header which is 20 bytes from the beginning of the packet. Subsequent Message Blocks will begin after the last byte of the previous Message Block.

Downstream Packet Message Block

Field Name	Offset	Len	Value	Notes
Message Length	*	2	NUM	Indicates the length in bytes of the message contained in this Message Block.
Message Data	*	*	ANUM	The message data.

* = Variable values

Message Length

The Message Length is an unsigned binary count representing the number of message data bytes following this Message Length field. The message length field value does not include the two bytes occupied by the message length field. The total size of the message block is the value of the message length field plus two.

Message Data

The Message Data is actual data of the message being transmitted by MoldUDP64. It is variable length and can be zero length. The meaning of the data is application specific.

Heartbeats

Heartbeats are sent periodically by the server so receivers can sense packet loss even during times of low traffic. Typically, these packets are transmitted once per second and contain the next expected Sequence Number. A Heartbeat packet is a MoldUDP64 packet with a Message Count of zero.

End of Session

When the current session is complete, Downstream Packets are sent with a Message Count of 0xFFFF(hex, or 65535 in decimal) for a short while in place of Heartbeats. These Downstream Packets contain the next expected Sequence Number, just like Heartbeats. While the End of Session messages persist, re-requests may be made on the current session. This is the last chance to ensure that all messages have been received.

Request Packet

The Request Packet is sent to request the retransmission of a particular message or group of messages. The request packet is sent to a Re-request server. A receiver may need to send this request when it detects a sequence number gap in received messages. The response to a valid Request Packet is a standard Downstream Packet unicast back to the source of the retransmission request. This allows downstream MoldUDP64 users to read the retransmitted Downstream Packet in their multicast processing socket if the request was made from that socket (in other words, the client need only have one socket open to listen to the multicast and to process retransmissions, even though the retransmissions are not multicast).

Request Packet

Field Name	Offset	Len	Value	Notes
Session	0	10	ANUM	Indicates the session to which this packet belongs.
Sequence Number	10	8	NUM	First requested sequence number.
Requested Message Count	18	2	NUM	The number of messages requested for retransmission.

Sequence Number

The Sequence Number field of the packet Header indicates the sequence number of the first message requested.

Requested Message Count

The Message Count indicates how many messages should be retransmitted. If the total size of the requested messages exceeds the maximum payload size of the of one UDP packet, only the number of messages that completely fit will be returned. Additional retransmission requests must be made for the subsequent messages if they are still desired.

Receiver Example

A typical MoldUDP64 receiver client would be configured with the following parameters:

- The UDP port to listen on and the Multicast group to join
- A list of one or more Request Servers that are available to answer retransmission requests for this stream. Each server is specified as a host IP address and a UDP port to which to send requests.
- A session and sequence number of the next expected message if the client is being restarted.

A typical MoldUDP64 receiver client might obey the following flowchart:

1. Open a UDP socket for the appropriate port and join the desired multicast group.
2. Examine the first received packet to determine the currently active session.
3. If the received session does not match the expected session, abort and report the error.
4. Examine the sequence number of the first recently received packet.
5. If the sequence number does not match the next expected sequence number, send a Request Packet to the Request Server with expected packet number. Wait for a new packet and return to step 4.
6. Process each of the received messages in the packet. If a Downstream Packet with the Message Count set to End of Session is received, handle the End of Session event.
7. Wait for a new packet and return to step 4.