UFO (UDP for Orders)

Version 1.0
July 3, 2008

1. Overview

UFO is a lightweight point-to-point protocol that allows delivery of a set of sequenced messages from a server to client in real-time using UDP/IP sockets. The protocol guarantees delivery of each sequenced message despite the possibility of transient network failures.

UFO is designed to be used in conjunction with higher level protocols that specify the content of messages transported by the protocol. The UFO protocol layer is opaque to the higher-level messages and may transport any message data, except each message must be short enough to fit within a single UDP/IP packet after protocol overhead.

Clients can send messages to the server. The UFO protocol layer provides no guarantees for these messages and they may encounter any possible failure of the underlying UDP/IP protocol such as packet loss. The UFO protocol does not acknowledge client messages in any way and defers this task to the higher-level server protocol.

UFO includes a simple scheme that allows the client to connect and authenticate with the server as a login mechanism.

1.1 Data Types

Alpha fields consist of alphanumeric ASCII characters, right padded with spaces where applicable.

Numeric fields are unsigned binary integers formatted in network byte order (big endian) with the most significant byte first.

Opaque fields are defined by a higher level protocol and consist of arbitrary data with no restrictions.

1.2 Protocol Description

Clients connect to a UFO server by first sending a Login Request message to the server. The server responds with a Login Accept packet if the login is successful. Otherwise, the server responds with a Login Reject packet. Since packets may be lost, the server always responds to each Login Request message, even if the client is already logged on.

Once connected, clients must periodically send Upstream Messages to the server. If the server does not receive an Upstream Message from the client for more than 10 seconds, the server disconnects the client and resumes listening for Login Request messages. Clients may send Heartbeat messages if no other messages are available to send. Clients may logoff explicitly by sending a Logoff Request message.

While the client is connected, all communications occur using the UDP/IP source address/port of the Upstream Packet containing the Login Request message. The server sends all Downstream Packets to this address/port pair, and discards received Upstream Packets that do not originate from this address/port pair. Once the client disconnects, Login Requests will be processed from any source address/port pair.

While the client is connected, the server sends Sequenced Data packets to the client in real-time as data is generated by the server. No flow-control is applied to these packets and all messages are always sent immediately by the server. If no data is available to send, the server will periodically send empty Sequenced Data packets that serve as heartbeat packets. These packets contain the sequence number of the next sequenced message to be transmitted. If the client receives no packets for more than 10 seconds, it should consider itself disconnected and attempt to login again.

Since Sequenced Data packets may be lost, clients may request retransmission of any sequenced messages by sending a Retransmission Request message to the server. The server processes each request and sends a Sequenced Data packet in response. If more messages are requested than fit into a single Sequenced Data Packet, only the number of messages that fit completely will be retransmitted.

The client may send Unsequenced Data messages to the server while it is connected to communicate with the higher-level server protocol. These messages are not acknowledged by the UFO protocol layer. The client is responsible for processing server sequenced data messages to determine whether any unsequenced data messages may have been lost.

1.3 Session Management

The server groups a related set of sequenced messages into a session. Each session has a unique ID that distinguishes it from all other sessions. The first sequenced message in each session has sequence number 1. Subsequent sequenced messages are numbered with sequentially incrementing sequence numbers. When connecting to a new session, clients must send Retransmission Request messages to request retransmission of any sequenced messages that may have been elapsed while the client was not connected.

Client connections persist for a single session only. The client is always disconnected before a new session begins.

When the current session is terminated, the server transmits an End of Session packet to the client to indicate that the session has ended and includes the total number of sequenced messages transmitted in the session. The server will not transmit additional sequenced messages beyond this count, though some previously transmitted packets may remain in the network. After session termination, the server retransmits the End of Session packet as a heartbeat instead of empty Sequenced Data packets. Unsequenced Messages transmitted by the client are ignored after End of Session. Clients may remain connected and request retransmissions after End of Session until the client times out, explicitly logs off, or is disconnected during shutdown. Once disconnected, client Login Request messages will be rejected or ignored until the next session begins.

Clients should send a blank requested session ID in the Login Request message when connecting to a new session. The server sends the current session ID as part of the Login Accept packet. Subsequent Login Request messages for the same session should contain the session ID the server sent in its Login Accept packet. If the session has ended, the server will reply to this explicit session request with a Login Reject packet indicating the requested session is invalid. In this case, the client should clear its session and request a new session using a blank session ID.

1.4 Limits

UFO assumes a 1500 byte network MTU, a 20 byte IPv4 header, and an 8 byte UDP header. As a result, the maximum UFO packet size is 1472 bytes. Sequenced and unsequenced messages must be small enough to not exceed this limit once protocol overhead is added.

2. UFO Downstream Packets

Each packet from the server to the client consists of a header containing a single byte that describes the type of the packet. The format of the remainder of the packet depends upon this type byte. The following types are defined:

Type	Description
A	Login accept
J	Login reject
S	Sequenced data
E	End of session

2.1 Login Accept Packet

The server sends a Login Accept packet in response to a valid Login Request message from the client. The server always responds with a Login Accept message for a valid Login Request, even if the client has already logged on.

Field Name	Offset	Length	Value	Notes
Packet type	0	1	Alpha	"A"
Session	1	10	Alpha	Current Session ID
Sequence Number	11	4	Numeric	Next message sequence number

2.2 Login Reject Packet

The server sends a Login Reject packet in response to an invalid Login Request message from the client.

Field Name	Offset	Length	Value	Notes
Packet type	0	1	Alpha	"J"
Reason code	1	1	Alpha	See Login Reject Codes below

Login Reject Codes:

Code	Description
A	Not authorized. The combination of username and password sent in the Login Request is not valid.
S	Invalid session. Either the requested session is invalid or not available.

2.3 Sequenced Data Packet

The server sends a Sequenced Data packet when sequenced data is available to send. If no data is available to send, the server sends a Sequenced Data packet containing zero messages to serve as a heartbeat packet.

The Sequenced Data packet consists of a Sequenced Data Header followed by zero or more Sequenced Message Blocks.

2.3.1 Sequenced Message Header

Field Name	Offset	Length	Value	Notes
Packet type	0	1	Alpha	"S"
Sequence number	1	4	Numeric	The sequence number of the first message within this packet.
Count	5	2	Numeric	The count of messages in this packet.

Sequence Number

The sequence number indicates the sequence number of the first message in this packet. If this is a heartbeat packet, this sequence number is the sequence number of the next message to be transmitted. If there are multiple Sequenced Message Blocks in this packet, each message following the first is implicitly numbered sequentially.

Message Count

The number of Sequenced Message Blocks that follow the Sequenced Data Header is specified by the Message Count field. If this packet is a heartbeat, this field contains zero and no Sequenced Message Blocks follow the Sequenced Data Header.

2.3.2 Sequenced Message Block

Field Name	Offset	Length	Value	Notes
Message Length	0	2	Numeric	Indicates the length in bytes of the message contained in this Sequenced Message Block
Message Data	2	Variable	Opaque	The contents of the sequenced data message

Message Length

The message length is an unsigned binary count in network byte order representing the number of message data bytes following the Message Length field.

Message Data

The Message Data field contains the actual content of the sequenced data message being transmitted by the UFO protocol. It is variable length and may contain any data byte values. This field may contain no more than 1465 bytes due to the packet size limitation. The meaning of the actual data is defined by higher-level application protocols.

2.4 End of Session Packet

The server periodically sends End of Session packets after the current session is terminated. The Sequenced Message Count field indicates the total number of sequenced messages that were transmitted in this session. This count is numerically one less than the next expected sequenced number as is transmitted in the heartbeat Sequenced Data packets.

Field Name	Offset	Length	Value	Notes
Packet Type	0	1	Alpha	"E"
Sequenced Message Count	1	4	Numeric	The total number of sequenced messages transmitted in this session

3. UFO Upstream Packets

Clients send messages to the server in the form of Upstream Packets that contain one or more Upstream Messages. The Upstream Packet consists of one or more Upstream Message Blocks. No header precedes the first Upstream Message Block. The count of Upstream Message Blocks is implied by the UDP/IP packet length. The end of the last Upstream Message Block in the packet must coincide with the end of the UDP/IP packet. Packets must contain at least one Upstream Message Block. A packet containing zero messages is not valid.

3.1 Upstream Message Block

Field Name	Offset	Length	Value	Notes
Message Length	0	2	Numeric	Indicates the length in bytes of the message contained in this Upstream Message Block
Message Data	2	Variable	Opaque	The contents of the upstream message

Message Length

The message length is an unsigned binary count in network byte order representing the number of Message Data bytes following the Message Length field. This field must be greater than zero.

Message Data

The Message Data field contains the actual content of the upstream data message. It is variable length and may contain any data byte values. Each message must contain no more than 1470 bytes due to the packet size limitation. Each message must be a valid Upstream Message as specified below.

3.2 UFO Upstream Messages

The client sends upstream messages to login, logoff, request retransmissions, send application messages, and to send heartbeats. Each client message starts with a header that consists of a single byte that identifies the type of the following message data. The following message types are defined:

Type	Description
L	Login Request
T	Retransmission Request
U	Unsequenced Message
R	Heartbeat Message
O	Logoff Request

3.2.1 Login Request Message

The client sends a Login Request message to connect to the UFO server. The client must send this message before sending any other messages to the server.

The client and server must have mutually agreed upon the username and password fields. They provide a simple authentication to prevent a client from inadvertently connecting to the wrong server.

Both the username and password are case-insensitive and should be padded on the right with spaces.

If the login is successful, the server replies with a Login Accept packet. If the login is not successful, the server replies with a Login Reject packet. The client should retransmit the Login Request message if no response is received.

Field Name	Offset	Length	Value	Notes
Message type	0	1	Alpha	"L"
Username	1	6	Alpha	Client username
Password	7	10	Alpha	Client password
Requested Session	17	10	Alpha	Specifies the session the client would like to log into, or all spaces to log into the currently active session.

3.2.2 Retransmission Request

The client sends a Retransmission Request Message to request retransmission of previously transmitted Sequenced Message Blocks. The request contains the sequence number of the first message to retransmit and the count of messages to retransmit. If more messages are requested than fit into a single Sequenced Data Packet, only the number of messages that fit completely will be retransmitted.

Field Name	Offset	Length	Value	Notes
Message type	0	1	Alpha	"T"
Sequence number	1	4	Numeric	First sequence number requested
Message count	5	2	Numeric	Number of messages requested

3.2.3 Unsequenced Message

The client sends Unsequenced Messages to interact with the higher level protocol transported by the UFO protocol. The content of the Message Data contained within the Unsequenced Message is defined by the higher level protocol. Since the packets that contain unsequenced messages may be lost, the client must be prepared to retransmit the Unsequenced Messages for important client events. The maximum length of an unsequenced Message Data payload is 1469 bytes due to packet size limitations.

Field Name	Offset	Length	Value	Notes
Message type	0	1	Alpha	"U"
Message data	1	Variable	Opaque	Unsequenced message defined by higher level protocol

3.2.4 Heartbeat Message

The client sends Heartbeat Messages to keep the connection alive when no other data is to be transmitted to the server. The Heartbeat Message contains no data other than the message type byte.

Field Name	Offset	Length	Value	Notes
Message type	0	1	Alpha	"R"

3.2.5 Logoff Request

The client sends a Logoff Request message to gracefully terminate its connection to the server. The server disconnects the client and does not respond to the Logoff Request message. The Logoff Request Message contains no data other than the message type byte.

Field Name	Offset	Length	Value	Notes
Message type	0	1	Alpha	"O"

4. Support

Questions about the UFO protocol specification may be sent to devsupport@nasdaqomx.com

5. Version History

Version	Date	Description

1.0 July 3, 2008 Initial distribution