

Lightspeed Gateway: Short Availability

Overview

The Short Availability Data service disseminates information describing the list of securities that are eligible for short sale. This service replaces any static borrowable or non-borrowable files previously published to disseminate short sale restrictions. The state of each security is distributed and updated in realtime as updates occur.

Client applications receive the current state of each security as well as proceeding updates by subscribing to symbols. During subscription, a snapshot of present data is disseminated. The end of this snapshot is marked by a specific message. Updates will be distributed to all client applications subscribing to the updated security.

Architecture

This service utilizes text-based line-oriented communications over standard TCP/IP sockets. Each TCP/IP connection to the server forms an independent session. Optional heartbeat messages may be utilized by the client to ensure integrity. If transmissions queue at the server, control messages indicating the beginning and end of queuing will be transmitted to indicate this to the client. If the client queue accumulates beyond reasonable limits, the connection will be terminated by the server.

All messages inbound and outbound are unsequenced and are the delimited by the end of the line. Thus each line carries precisely one message. Lines are terminated by carriage-return line-feed pairs (ASCII 0x0D and ASCII 0x0A respectively).

Fields within each message are delimited by single space characters (ASCII 0x20). Fields are identified by position within each message. Each message contains as its first field a type code. The meaning of the remaining fields is determined by this code.

The message type code is a two-character string determining the message type. The first character represents the message category and the second character identifies the message type within the message category.

Client applications should ignore unknown message types and unknown fields. These messages should be handled gracefully in such a manner as to not affect normal processing.



Client applications may transmit messages at any time to make requests or effect control actions. Requests received concurrently may be processed concurrently.

Data Types

All data and message fields are represented in plain ASCII form. There are no special data types or encodings used in this specification.

Message Types

Messages within are broken into two categories: control and short availability data. The message categories and types are summarized below.

Message categories:

Category Code	Message Category
_	Control
Н	Short Availability Data

The following messages are defined as control messages:

Category Code and Type Code	Direction ¹	Message Description
_H	Inbound	Heartbeat request
_h	Outbound	Heartbeat response
Q	Outbound	Queuing begins
_q	Outbound	Queuing ends

The following messages are defined as short availability data messages:

Category Code and Type Code	Direction	Message Description
HS	Inbound	Subscribe
HQ	Inbound	End subscription
HS	Outbound	End of snapshot marker
HU	Outbound	Short availability update

¹ Inbound is from client to server, outbound is from server to client.



Control Messages

The following messages are used for control purposes between the server and client applications.

Heartbeat Request Type `_H` Inbound to Server

¹ Inbound is from client to server, outbound is from server to client.

Sent to solicit a heartbeat response message. This message can be used in conjunction with the Heartbeat Response to detect underlying TCP/IP connection problems.

This message consists of only the type field.

Field Number	Description
1	Message Type, text `_H`

Heartbeat Response

Type `_h` Outbound to Client Application

Sent to acknowledge a previous heartbeat request. This message can be used in conjunction with the Heartbeat Request to detect underlying TCP/IP connection problems.

This message consists of only the type field.

Field Number	Description
1	Message Type, text `_h`

Queuing Begins

Type \underline{Q} Outbound to Client Application



Sent by the server to the client application to indicate that data is being queued at the server. The client may wish to react to this condition.

This message consists of only the type field.

Field Number	Description
1	Message Type, text `_Q`

Queuing Ends

Type `_q` Outbound to Client Application

Sent by the server to the client application to indicate that data is no longer being queued at the server. The client may wish to react to this condition.

This message consists of only the type field.

Field	Description
Number	
1	Message Type, text `_q`

Short Availability Data Messages

Subscribe

Type `HS` Inbound to Server

Sent to initiate a subscription to a given symbol or set of symbols described by a wildcard pattern.



	Description	
Number		
1	Message Type, text `HS`	
2	Symbol or wildcard pattern	

The server responds to this message with a snapshot of newly subscribed data. This snapshot is in the form of a sequence of Short Availability Update messages followed by an End of Snapshot Marker. This snapshot only contains newly subscribed symbols. Symbols that match two or more subscription patterns will be transmitted only once. See Subscription Notes below for more information.

See Wildcard Syntax below for a description of the wildcard pattern syntax.

Subscription	
Type `HQ` Inbound to Server	

Sent to terminate a subscription to a given symbol or set of symbols described by a wildcard pattern. The pattern must match a pattern previously subscribed in a Subscribe message.

Field Number	Description	
1	Message Type, text `HQ`	
2	Symbol or wildcard pattern	

The server does not respond to this message.

End of Snapshot Marker

Type `HS` Outbound to Client Application

Sent to indicate the end of a data snapshot. This message concludes a block of updates for the subscription. The pattern matches the pattern provided in a previous Subscribe message.



Field Number	Description
1	Message Type, text `HS`
2	Symbol or wildcard pattern from Subscribe message

When processing several Subscribe messages at the same time, the server may transmit a block of End of Snapshot messages for each of the processed subscriptions following the appropriate snapshot data. Thus this message does not delimit snapshot data across subscriptions, and it only indicates that no further snapshot data will be transmitted for the given subscription pattern.

Short Availability Update

Type `HU` Outbound to Client Application

Sent to indicate an update to the availability of a given security for short sale.

Field Number	Descripti	on	
1	Message Type, text `HU`		
2	Security s	symbol	
3	Short availability flag. This field takes the following values:		
	Value Y	Description Available for short sale	
	H Hard to borrow – available for short sale when locate is available.		
	X Not available for short sale		
	Т	Threshold Security	
	N	Unknown – Call for Locate	



Wildcard Syntax

When subscribing to symbols, client applications may utilize wildcards to indicate a set of desired symbols. These wildcards are described below.

A '*' character will match zero or more characters.

A '?' character will match one character.

The expression '[...]' matches one character from the set of characters between the brackets. If the first character of the set is a '!' character, the wildcard is complemented and matches one character not in the set. A sequence of characters may be abbreviated using a '-' character.

Examples:				
*	Matches all symbols			
*Z	Matches any symbol that ends with Z			
[ABC]*	Matches any symbol beginning with A, B, or C			

Note that the wildcards will match symbols that do not yet exist at the time of subscription. Therefore a subscription to '*' will provide all available data.

Subscription Notes

The wildcard pattern allows subscription to more than one symbol with one Subscribe message. Typically, a subscription to '*' will be performed to acquire all data.

A subscription pattern will subscribe to any currently known symbols that match the pattern as well as any symbols generated after the subscription that also match the pattern. Thus is it possible to receive updates for new symbols not represented in a prior snapshot.

Since patterns may match more than one symbol, it is possible for two patterns to match the same symbol. The server permits such overlapping subscriptions. The server handles these overlaps exactly as the following describes:

1. When processing a Subscribe message, the server responds with a snapshot for any given symbol exactly once, no matter how many subscribed patterns match that symbol. That is, a subscription returns snapshots only for newly matched symbols.



2. The End Subscription message terminates updates for a given symbol once all patterns that match the symbol are terminated via an End Subscription message. If a new subscription is requested for the symbol via a Subscribe message, then a new snapshot will be sent in accordance with note 1.

To summarize, Subscribe and End Subscription messages will ignore any symbol that matches another outstanding subscription pattern.

Example:

- 1. Client connects
- 2. Client subscribes to ABC
- 3. Server transmits snapshot for ABC
- 4. Client subscribes to AB*
- 5. Server transmits snapshot for ABA, ABB, ABD, etc. but not ABC
- 6. Client ends subscription to ABC
- 7. Server continues to disseminate updates about ABC as this symbol still matches the subscribed AB* pattern
- 8. Client subscribes to ABC
- 9. Server transmits no snapshot for ABC (but still transmits End of Snapshot marker)
- 10. Client ends subscription to AB*
- 11. Server discontinues updates for all symbols except ABC
- 12. Client ends subscription to ABC
- 13. Server discontinues all updates

Sample Messages

The following is a sample communication between a client application and a short availability server.

Messages to the server from the client application are indicated with a ' \leftarrow '. Messages from the server to the client application are indicated with a ' \rightarrow '.

Direction	Message	Comment
\leftarrow	HS AAPL	Client Subscribes to AAPL
\rightarrow	HU AAPL X	Server sends snapshot for AAPL
\rightarrow	HS AAPL	End of snapshot for AAPL
\rightarrow	HU AAPL H	Update for AAPL arrives
\rightarrow	HU AAPL Y	Update for AAPL arrives

Copyright © 2009 Lightspeed Trading, LLC. Proprietary Information. All rights reserved



←	HS A*	Client Subscribes to all symbols beginning with A
\rightarrow	HU AAA X	Server sends snapshot, updating AAA and ABC
\rightarrow	HU ABC H	
\rightarrow	HS A*	End of snapshot for A*
\rightarrow	HU AMGN Y	Update for a new symbol matching A*
\rightarrow	HU AAA N	Update a previously subscribed symbol
←	HQ A*	Client Ends Subscription to A*
\leftarrow	HQ AAPL	Client Ends Subscription to AAPL

Revision History

Version	Date	Author	Description
Draft 1	09 Mar 2004	revans	Initial version of this document.
Draft 2	09 Mar 2004	revans	Added Sample Messages section.
Version 1.0	10 Mar 2004	revans	Added connection details to Architecture section. Added details regarding multiple subscriptions to End of Snapshot message description.
Version 1.01	28 Jan 2005	slowenbe	Added new T (threshold) and I(Intraday-only) availability values.
Version 1.02	17 Mar 2007	Ysharon	Changed categories – removed I, added H (Hard to borrow)
Version 1.03	04 May 2007	RFalcon	Removed 'D Negative Rebate' short availability flag
Version 1.04	06 Jan 2009	LCapone	Renamed Lightspeed Gateway Short Availability.