

Citrius - Book Server protocol

version: 1.0.a

revision date: Jan 14, 2025

Changes with revision 1.0.a		
date	breaking change	change
2.27.23	NO	selective catalog request (via payload) is now supported. Removed note to the contrary under catalog request message description.
3.16.23	NO	added "market independent SIP" to participant id enumeration
3.17.23	NO	corrected label of participant id bzx (was incorrectly spelled byz)
3.29.23	NO	adjust participant id enum to match change in citrius_multicast spec
4.17.23	YES	Added quote condition indicator book_response message
8.9.23	YES	fixed error in documentation of book_response. reserved field should read char[29] and not char[30]
8.9.13	YES	added message size field to message header
8.11.13	NO	fixed binary examples of messages
8.11.13	YES	Fixed ordering of fields in catalog_request message
8.15.23	NO	fixed error in documentation of book_status message
9.27.23	YES	updated NBBO indicator
9.27.23	NO	added error codes: <ul style="list-style-type: none">• no book specified• invalid market feed id• invalid feed type
10.16.23	YES	Add fields to book_status message: <ul style="list-style-type: none">• last trade settlement type indicator• last trade trade through exemption indicator• extended hours indicator• sro trade indicator• trade specific properties Reduced 'reserved' field by 5 bytes
10.16.23	YES	Added fields to book_response message: <ul style="list-style-type: none">• order id• mpid Reduced 'reserved' field by 12 bytes
2.2.24	YES	increased book status volume field to 64 bits
2.5.24	YES	increase size of market feed id to 8 bytes
2.21.24	YES	added symbol name field to book_status message
3.1.24	YES	added 'unsorted order' to book_format
3.5.24	YES	added feed_format and replaced book_format field in multicast_description, catalog request messages
7.23.24	NO	Added edga, edgx, bats, nyse, nyse_arca to quote source enumeration
7.23.24	NO	Added trades to feed_format enumeration

1.14.25	NO	Merge enumerations “book format” and “feed format”. Remove “book format” and add “imbalance” and “reserved” to “feed format”
---------	----	---

data types:

- integers
- enumerations
- timestamps
- prices
- book identifier
- ip address

message structure:

- message header
- message type

RPC structure:

- RPC request header
- RPC response header

messages:

- login request
- login response
- multicast description
- catalog request
- catalog response
- book request
- book status
- book response
- book complete
- error response

Enumerations:

- RPC response status
- message type
- buy sell indicator
- license status
- error code
- quote source
- financial instrument
- feed format
- NBBO status indicator
- trading status indicator
- short sale restriction indicator
- limit up-limit down indicator
- participant id
- extended hours indicator
- sro trade indicator
- settlement type indicator
- trade through exemption indicator

The Citrius Book Server Protocol is used to communicate with the Lime Citrius 2.0 live book snapshot service. It is an RPC based protocol over TCP and supports asynchronous multipart responses.

Endian: All integer values are represented in big endian format.

Sequence Numbers:

Each multicast datagram will contain exactly one block of multicast messages and is uniquely identified by a sequence number. This sequence number will increase by the number of messages contained within each multicast datagram. Gaps in the multicast feed are therefore detected via missing sequence numbers.

When a multicast stream's block sequence number reaches the maximum 64 bit unsigned integer value of 18446744073709551615 it will wrap to the next expected sequence number which will be 0.

Multicast Source Identifiers:

Each block header contains a source identifier field which will remain constant for all blocks which originate from the same instance of the same Citrius multicast server. The source identifier is the number of milliseconds since midnight at which time the particular instance of the Citrius multicast server was created. Therefore, failover/restarts of Citrius multicast servers can be detected by a change in the source identifier field. The sequence number which appears with each unique source identifier will be reset to start at 1. In the event that a multicast stream momentarily contains datagrams with varying source identifier values only those with the larger of the two source identifiers should be considered valid as this field represents the time when the originating instance of a Citrius multicast server was started and, therefore, datagrams with lower source identifier field should be regarded as 'stale' and as having originated from an instance which has been superseded by a more recent instance.

Data Types:

Integer types are unsigned unless otherwise specified. The size of integer fields is as follows:

Name:	Size:	Range:
integer_8	8 bits	[0 -> 255]
integer_16	16 bits	[0 -> 65,535]
integer_32	32 bits	[0 -> 4,294,967,295]
integer_64	64 bits	[0 -> 18,446,744,073,709,551,615]

Enumerated values are unsigned integer fields. The size of enumerated fields is as follows:

Name:	Size:	Range:
enum_8	8 bits	[0 -> 255]
enum_16	16 bits	[0 -> 65,535]

Timestamps:

Timestamps are 8 bytes in size (integer_64) and represent the nanosecond since epoch.

Prices:

Price fields are 4 bytes (integer_32) or 8 bytes (integer_64) in size and are followed by a one byte field indicating the number of decimal places in the primary Integer price field.

Name:	Size:	Range:
price_32	integer_32	[0 -> 4,294,967,295]
price_64	integer_64	[0 -> 18,446,744,073,709,551,615]

example of price_32 structure: struct price_32 { integer_32 mantissa; integer_8 exponent; };	example of price_64 structure: struct price_64 { integer_64 mantissa; integer_8 exponent; };
--	--

Book Identifiers:

The book for each unique symbol or OSI is referenced by a unique 64 bit (integer_64) symbol mapping.

IP Address:

An IP address is six bytes long and takes the form:

IP Address		
Field type	Field Description	Offset
integer_32	IPv4 network identifier	0
integer_16	Port identifier	4
Total Size: 6 Byte		

Message structure:

Each Citrius message begins with the standard message header structure:

message header		
Field type	Field Description	Offset
message type	indicates the type of message which follows	0
integer_16	message size	1
		Total Size: 3 Byte

RPC structure:

This protocol is RPC based. Responses are asynchronous and consist of a single response or be constructed of multiple response messages. After the standard message header, each request message continues with a single RPC request header:

RPC request header		
Field type	Field Description	Offset
integer_32	unique request identifier	0
		Total Size: 4 Bytes

Similarly, for response messages, after the standard message header, each response message continues with a single RPC response header:

RPC response header		
Field type	Field Description	Offset
integer_32	unique request identifier	0
RPC response status	indicates if end of response or if a multipart response	4
		Total Size: 5 Bytes

Log in - Connecting with a Citrius Book Server:

The Citrius Book Server is designed to be low fault tolerant with regards to the login process and will immediately close the TCP connection if any of the following rules are not met.

- The first message sent on a newly established TCP connection with a Citrius Book Server must be a login request message.
- No subsequent data should be transmitted after sending a login request message until such time as the login response message has been received.
- Login failures will result in the TCP connection being terminated immediately after the login response message is sent.

Citrius 2.0 Licenses are per unique IP address and are valid for the entirety of that day. Establishing a connection with a Citrius Book Server will consume one Citrius License for the entirety of the day regardless of the duration of that TCP connection. Citrius Licenses are associated with the IP Address of the client which has established the connection. Therefore, subsequent logins/connections from the same client IP address **do not** consume additional Citrius Licenses.

Example: A client has two Citrius Licenses and connects to a Citrius Book Server from IP address 1.2.3.4. This will consume one of the client's Citrius Licenses. The client then disconnects from the Citrius Book Server and will still have only one remaining Citrius License for that day. The client then connects with a Citrius Book Server from IP address 3.4.5.6 and the second Citrius License is then consumed. The client now has zero remaining open Citrius Licenses for that day. If the client then attempts to connect from IP address 1.2.3.4 the connection will not require an additional license as the client has already established a Citrius License from that source IP address on that day. However, a subsequent attempt to connect from IP address 5.6.7.8 will be rejected as the client has no more available Citrius Licenses for the day.

Upon successful validation of the client's credentials as provided with the login request message, the Citrius Book Server will send both a login response message as well as a series of multicast description messages which detail which Citrius Multicasts are available via that Citrius Book Server.

The following is the proper exchange of messages for establishing a connection with a Citrius Book Server:

1. Client sends a login request message with valid account name and password.
2. The client does not send any further data to avoid a connection close.
3. The Citrius Book Server sends a login response message with a license status code of 'valid'.
4. The Citrius Book Server sends additional multicast description messages detailing which Citrius Multicasts this Citrius Book Server is servicing. The final multicast description message will have an RPC status of 'complete' which indicates that the login process has been completed.
5. The client may then send additional requests as desired.

Messages:

login request		
Field type	Field Description	Offset
message header	message type = 8, message length = 263	0
RPC request header	user defined call id	3
char[128]	account name	7
char[128]	password	135
Total Size: 263 Bytes		

login response		
Field type	Field Description	Offset
message header	message type = 9, message size = 12	0
RPC response header	RPC id and response status	3
license status	response code	8
integer_8	book server major version	9
integer_8	book server minor version	10
char	book server version letter	11
Total Size: 12 Bytes		

multicast description		
Field type	Field Description	Offset
message header	message type = 10, message size = 51	0
RPC response header	RPC id and response status	3
quote source	original source of quote data	8
financial instrument	options, equities, etc	9
feed format	type of feed represented by this multicast	10
integer_64	unique identifier for this Citrius multicast feed	11
IP address	the A side IP address for this Citrius multicast feed	19
IP address	the B side IP address for this Citrius multicast feed	25
integer_64	book id cipher	31
integer_64	price id cipher	39
integer_64	size id cipher	47
Total Size: 55 Bytes		

The Multicast Description message provides information about each of the Citrius Multicast streams that the particular Citrius Book Server services. This information includes the origin of the multicast stream, the type of financial instrument and the feed format which the multicast describes. Additionally, if the Citrius Multicast has “field obfuscation” enabled, the three cipher fields will be set to non-zero values. These value must be XOR-ed with any multicast message’s book_id, price and size/volume fields in order to interpret those fields correctly. Currently, field obfuscation is not enabled on any Citrius servers.

RPC ID = 1, RPC status = *incomplete*, Quote Source = *OPRA*, Instrument = *Options*, Feed Format = *national consolidated bid and best offer*, Citrus market feed id = 1, "A" side IP address = 0.1.2.3:4, "B" side IP address = 5.6.7.8:9, book id cipher = 0x0123456789abcdef, price field cipher of 0xabcdef0123456789, size/volume cipher of 0x0a1b2c3d4e5f6789.

Getting Book IDs:

The Citrius Book Server is used to attain the book id (symbol mappings) for the symbols which appear on the Citrius Multicasts. To request the book id for any given symbol the client should send a catalog request message and provide a payload containing the symbols of interest. If no payload is provided then the request is considered a request for the entire universe of symbols. The client must also provide the quote source, feed format and financial instrument type in order to specify which of the available multicast feed types the client is interested in.

In the case where financial instrument is set to "options" the list of symbol names can be used to specify the underlying instrument and the catalog response will contain multiple OSI symbol mappings - one for each OSI symbol which shares that underlying instrument.

catalog request		
Field type	Field Description	Offset
message header	message type = 34, message size = 12+	0
RPC request header	user defined call id	3
integer_16	payload size N (<i>Max</i> = 512)	7
quote source	original source of data	9
financial instrument	options, equities, etc	10
feed format	feed type	11
char[512]	optional list of symbols to filter response	12
Total Size: 12 Bytes + N		

Example of a catalog request message for:

RPC ID = 3, Quote Source = SIAC, Instrument = Equities, Feed Format = consolidated bid and best offer, symbols F and AA.

0x22	0x00	0x10	0x00	0x00	0x00	0x03	0x00	0x04	0x02	0x01	0x03	0x46	0x20	0x41	0x41
Message Type	Message Length		RPC ID				payload size		SIAC	equities	consolidated BBO	F	[space]	A	A

Each catalog request message will receive one or more catalog response messages and those response messages shall have payloads containing book descriptions (symbol name, book id and the id of the Citrus multicast which carries message for that symbol) indicating the symbol mapping for each symbol requested. If a symbol was not found for the specified multicast feed properties then a book id of zero shall be provided to indicate this fact.

catalog response		
Field type	Field Description	Offset
message header	message type = 35, message size = 10+(44N)	0
RPC response header	RPC id and response status	3
integer_16	payload arity N (<i>Max = 128</i>)	8
book description[]	array of book names and book identifiers	10
		Total Size: 10 Bytes + (44N)

book description (payload type for catalog response message)		
Field type	Field Description	Offset
char[32]	symbol/OSI name	0
book id	book id associated with symbol/OSI name	32
integer_64	id of Citrius multicast feed where symbol appears	40
		Total Size: 48 Bytes

Example of a catalog response message for the above catalog request message:

0x23	0x00	0x62	0x00	0x00	0x00	0x03	0x02	0x00	0x02	0x46	0x20	0x20	0x20	0x20	0x20
Message Type	Message Length		RPC ID				RPC Status	Payload Length		Symbol name "F" ...					
0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20
... Symbol name "F" ...															
0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0xae	0xbc	0x3f	0xde	0x55	0x46
... Symbol name "F"										Book ID for "F" ...					
0x41	0x5a	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x41	0x41	0x20	0x20	0x20	0x20
... Book ID for "F"		Citrus Feed ID for "F"								Symbol name "AA" ...					
0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20
... Symbol name "AA" ...															
0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x20	0x2a	0xf3	0x96	0x4d	0x7c	0x82
... Symbol name "AA"										Book ID for "AA"					
0x2f	0x33	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01						
Book ID for "AA"		Citrus Feed ID for "AA"													

Book snapshots:

After attaining book ids using the catalog request message, the client can request current book snapshots using those book ids. To do this the client should use the book request message. The request must specify the id of the Citrius multicast where the book appears (according to the catalog response information) and, optionally, can request that the book snapshot only be delivered at or after the sequence number provided in the request. This optional feature can be used by client's to more easily sync the live Citrius multicast data with the book snapshot attained by the book request message*.

book request		
Field type	Field Description	Offset
message header	message type = 15, message length = 29 + (8N)	0
RPC request header	user defined call id	3
integer_16	payload arity N (Max = 256)	7
integer_64	id of Citrius multicast feed where symbol appears	9
source identifier	(optional*) snapshot only if feed is same source identifier	17
sequence number	(optional*) snapshot after this number if provided	21
book id[N]	payload - list of book identifiers	29
Total Size: 29 Bytes + (8N)		

* This feature is not currently supported but will be in a future release.

For each book requested a series of response messages will be generated - beginning with a single book_status response, followed by one or more book_response messages (depending on the feed format associated with the requested Citrius feed id) and finally, a single book_complete message. The book_status message contains general information about the requested book including high/low and open price, last trade information, total volume, etc. Additionally, a book_status message contains a payload describing trading status, price bands, short sale restrictions etc. This payload may contain more than one entry if the associated multicast is a consolidated feed.

book status		
Field type	Field Description	Offset
message header	message type = 18, message length = 152 + (14N)	0
RPC response header	RPC id and response status	3
integer_16	payload arity N (Max = 64)	8
char[32]	symbol name / osi	10
book id	identifier of book being referenced	42
integer_64	Citrius feed id	50
source identifier	multicast source identifier	58
sequence number	multicast sequence number	62
integer_64	volume	70
price_32	high price mantissa	78
integer_8	high price exponent	82
price_32	low price mantissa	83
integer_8	low price exponent	87
price_32	open price mantissa	88
integer_8	open price exponent	92
price_32	last trade price mantissa	93
integer_8	last trade price exponent	97
integer_32	last trade size	98
timestamp	last trade timestamp	102
integer_64	last trade identifier	110
participant id	last trade participant id	118
settlement type indicator	last trade settlement type	119
trade through exemption indicator	last trade trade through exemption	120
extended hours indicator	last trade extended hours	121
sro trade indicator	last trade sro trade detail	122
integer_8	last trade feed specific properties	123
price_32	close price mantissa	124
integer_8	close price exponent	128
timestamp	close timestamp	129
char[12]	reserved	137
trading status indicator	trading status	149
short sale restriction indicator	short sale restriction	150
trading session	trading session indicator	151
book participant status[N]	payload - list of participant status for this book	152
Total Size: 153 Bytes + (14N)		

book participant status (payload type for book status message)		
Field type	Field Description	Offset
participant id	identifier for participant	0
price_32	limit upper price mantissa	1
integer_8	limit upper price exponent	5
price_32	limit lower price mantissa	6
integer_8	limit lower price exponent	10
LULD indicator	limit up/limit down indicator	11
trading status indicator	trading status for this participant	12
short sale restriction	short sale status for this participant	13
Total Size: 14 Bytes		

book response		
Field type	Field Description	Offset
message header	message type = 17, message length = 66	0
RPC response header	RPC id and response status	3
book id	identifier of book being referenced	8
buy sell indicator	indicates side	16
integer_32	size	17
price_32	price mantissa	21
integer_8	price exponent	25
participant id	origin of quote	26
timestamp	exchange timestamp	27
NBBO status indicator	indicates if quote is NBBO (<i>when applicable</i>)	35
quote condition indicator	quote condition	36
order id	exchange order id	37
char[4]	mpid	45
char[17]	reserved	49
Total Size: 66 Bytes		

book complete		
Field type	Field Description	Offset
message header	message type = 36, message size = 16	0
RPC response header	RPC id and response status	3
book id	identifier of book being referenced	8
Total Size: 16 Bytes		

error response		
Field type	Field Description	Offset
message header	message type = 37, message size = 9	0
RPC response header	RPC id and response status	3
error code	reason for error	8
Total Size: 9 Bytes		

Enumerations:

Various messages contain fields which reference enumerations. These enumerations are as follows:

RPC response status	enum_8
undefined	0
incomplete	1
complete	2
Total Size: 1 Byte	

message type	enum_8
<i>[undefined/reserved]</i>	0
log in request	8
log in response	9
multicast description	10
book request	15
book response	17
book status	18
book complete	36
catalog request	34
catalog response	35
error response	37
Total Size: 1 Byte	

buy sell indicator	enum_8
<i>undefined</i>	0
unknown	1
buy	2
sell	3
both	4
none	5
Total Size: 1 Byte	

license status	enum_8
<i>undefined</i>	0
invalid unknown account	1
invalid password	2
invalid not licensed for quote source	3
invalid not licensed for feed format	4
invalid not licensed for financial instrument	5
invalid not licensed for ip address	6
invalid license count exceeded	7
invalid unique ip address count exceeded	8
valid	9
invalid service not available	10
invalid quote source	11
invalid requested feed not available	12
Total Size: 1 Byte	

error code	enum_8
<i>undefined</i>	0
invalid request message received	1
invalid request prior to login completion	2
not logged in	3
already logged in	4
invalid session	5
no book specified	6
invalid market feed id	7
invalid feed type	8
Total Size: 1 Byte	

quote source	enum_8
<i>undefined</i>	0
opra	1
siac	2
tvitch	3
otc	4
citrius	5
utp	8
edga	9
edgx	10
bats	11
nyse	12
nyse arca	13
Total Size: 1 Byte	

financial instrument	enum_8
<i>undefined</i>	0
equities	1
options	2
Total Size: 1 Byte	

feed format	enum_8
<i>undefined</i>	0
national consolidated bid and best offer	1
depth of book	2
consolidated bid and best offer	3
bid and best offer	4
trades	5
imbalance	6
reserved	7
Total Size: 1 Byte	

NBBO status indicator	enum_8
Description	Bit flag
<i>undefined</i>	0
no nbbo bid	0x01
no change to nbbo bid	0x02
is nbbo bid	0x04
has nbbo bid appendage	0x08
no nbbo ask	0x10
no change to nbbo ask	0x20
is nbbo ask	0x40
has nbbo ask appendage	0x80
Total Size: 1 Byte	

NBBO status indicator is used to indicate if top of book messages contain NBBO bid and/or ask information and if such information is defined in top of book appendages which immediately follow top of book messages. If the NBBO status indicator is 0x08 or 0x80 then there will be exactly one such appendage message. If the NBBO status indicator is 0x88 then there will be exactly two such appendages with the bid appendage preceding the offer appendage.

trading status indicator	enum_8
<i>undefined</i>	0
trading halted	1
trading paused	2
trading	3
quotes only	4
unknown	5
revoked	6
deleted	7
security status not applicable	8
opening delay	9
no open no resume	10
price indication	11
trading range indication	12
market imbalance buy	13
market imbalance sell	14
market on close imbalance buy	15
market on close imbalance sell	16
reserved	17
no market imbalance	18
no market on close imbalance	19
Total Size: 1 Byte	

short sale restriction indicator	enum_8
Description	Value
<i>undefined</i>	0
short sale restriction activated	1
short sale restriction in effect	2
short sale restriction continued	3
short sale restriction not in effect	4
short sale restriction deactivated	5
unknown	6
Total Size: 1 Byte	

settlement type indicator	enum_8
Description	Value
<i>undefined</i>	0
regular	1
cash	2
next day	3
seller	4
yellow flag	5
Total Size: 1 Byte	

trade through exemption indicator	enum_8
Description	Value
<i>undefined</i>	0
no reason	1
inter market sweep	2
market center open	3
derivatively priced	4
reopening	5
market center close	6
qualified contingent	7
reserved	8
corrected consolidated close	9
Total Size: 1 Byte	

extended hours indicator	enum_8
Description	Value
<i>undefined</i>	0
not extended hours	1
form t	2
sold last	3
sold out of sequence	4
extended trading hours sold out of sequence	5
Total Size: 1 Byte	

sro trade indicator	enum_8
Description	Value
<i>undefined</i>	0
no sro required	1
average price	2
automatic execution	3
price variation	4
odd lot	5
rule 127	6
rule 155	7
market center close	8
prior reference price	9
market center open	10
contingent	11
cross trade	12
stopped stock	13
acquisition	14
bunched	15
distribution	16
placeholder	17
bunched sold trade	18
split trade	19
Total Size: 1 Byte	

limit up limit down indicator	enum_8
Description	Value
<i>undefined</i>	0
unknown	1
price band	2
republished price band	3
not applicable	4
Total Size: 1 Byte	

participant id	enum_8
Description	Value
<i>undefined</i>	0
unknown	1
nyse american	2
nyse national	3
nyse chicago	4
nyse	5
nyse arca	6
nasdaq gemx	7
<i>reserved/deprecated</i>	8
nasdaq mrx	9
nasdaq omx psx	10
nasdaq omx bx	11
nasdaq	12
nasdaq options market	13
nasdaq bx options	14
<i>reserved/deprecated</i>	15
finra alternative display facility	16
international securities exchange	17
long term stock exchange	18
members exchange	19
investors exchange	20
boston options exchange	21
miami international securities exchange	22
miax pearl	23
miax emerald	24
otc	25
cboe	26
cboe edga	27
cboe edgx	28
cboe byx	29
cboe bzx	30
cboe options exchange	31
cboe edgx options	32
cboe c2 options exchange	33
cboe bzx options exchange	34
consolidated quotation system	35
options price reporting authority	36
market independent sip	37
Total Size: 1 Byte	

