Nasdaq Smart Options

Product Overview

The Smart Options feed is a data feed product offered by Nasdaq, Inc. The Smart Options feed provides the National Best Bid and Offer (as calculated by the Options Price Reporting Authority - OPRA), Last Trade and various other instrument related information from OPRA.

Publisher

Nasdaq Global Information Services equips investors with the tools to make informed decisions by providing innovative real-time and historical analytic products and intelligent solutions designed to tap new data sets and meet new industry challenges. Nasdaq Data Link eliminates infrastructure and resource obstacles from collecting and storing large data sets, providing tools with industry adopted open standards to ingest and analyze market data and other financial information. To learn more about the company, technology solutions and career opportunities, visit us on LinkedIn, on Twitter @Nasdaq, or at www.nasdaq.com.

Delivery

Nasdaq Data Link provides a modern and efficient method of delivery for real-time exchange data and other financial information. Data is made available through a suite of APIs, allowing for effortless integration of data from disparate sources, and a dramatic reduction in time to market for customer-designed applications. The API is highly scalable, and robust enough to support the delivery of real-time exchange data.

This repository provides an SDK for developing applications to access the Nasdaq Data Link API. While the SDK is open source, connecting to the API does require credentials, which are provided by Nasdaq during an on-boarding process.

For more information please use the link- https://github.com/Nasdaq/CloudDataService

Table of Contents

1 Data Types:

Timestamps are formatted as numeric fields and express time as nanoseconds since midnight, Eastern Time (unless otherwise specified). Timestamps may be represented either as a single Long (8 bytes), or broken apart into two integers: one containing the whole number of seconds, and one containing the nanosecond offset from that whole number of seconds. For bandwidth efficiency, the market data timestamps received from the upstream source are typically broken apart, with the nanosecond portion included within the market data messages and the whole second portion disseminated in a separate Channel Seconds message when required.

Many of the market data messages have a Short, Long, and Extended Form. The field sizes in these forms are adapted to handle the typical needs of the market message, rather than the theoretical needs. If possible, the smallest form will be used in order to reduce bandwidth.
Many of the market data messages contain an Appendage List of zero or more appendage elements. These appendage lists are used to publish "*optionally available*" information alongside the common fields of the event. Consumers of the data are not required to process this information, but it will be provided if available, should the client prefer to utilize it. For example, End Of Day Summaries from OPRA provide High/Low/Last values for Options instruments. These will be communicated as appendages on a ValueUpdate message.

Option contracts are indirectly linked to their Underlying Symbols via the PARENT_LOCATE appendage type, which is normally provided on the Instrument Locate messages. In the morning, the Underlying Symbols (denoted with their particular Product Type codes – Equity, Index, etc.) will be spun out. The underlying symbols will be followed by Instrument Locate messages for the Option Roots (which are sent with a Product Type of "Option Root" – see Appendix). The Instrument Locate messages for the Option Roots will contain the PARENT_LOCATE appendage value code, with the value of the appendage set to the locate code for the Underlying Instrument. Any special settlement delivery components will be sent with a locate code of the Option ROOT (as opposed to the individual put/call/expiration/strikes). Once data is disseminated for an individual Option contract, the Symbol Locate will be disseminated with a Product Type of Option, and with a PARENT_LOCATE appendage linking it to the Option Root instrument. The conceptual layout for these records will then be Option -> Option Root -> Underlying. If an issue with the reference data is discovered that requires the linking between the Option Root and the Underlying to be re-published intraday, they will be sent appended to Instrument Meta Data messages for the Option Root. *It is important to note that Special Settlement Delivery Components and Underlying information is only published once for an Option Root, and not per individual contract expiration and strike price.*

2 Message Formats:

2.1 Data Appendage Lists

Appendage Lists are appended to the end of many market data messages. The list will contain zero or more appendage elements, and continue until the end of the message - no common message fields will follow the list. All appendage elements will have a standard appendage header consisting of an Unsigned Length value, which indicates the length of the appendage value (not including the common appendage header); a format code, which indicates how the value is formatted; a value type code, which indicates what the value represents; and a value portion, which is variable based on the format code. "*Pair*" appendages provide two values - the data represented by these values is determined by the Value Type Code. **Value Type Codes** are determined by the message category of the message containing the appendages. For example, InstrumentLocate messages have a different list of Value Type Codes than ValueUpdate messages.

2.1.1 Appendage Value Formats

2.1.1.1 Decimal Value – Short Form Appendage Element level heading

Short Form Decimal Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	2
Format Code	1	1	Numeric	1 – Short Form Decimal
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	2	Decimal	Implied 2 places of precision

2.1.1.1 Decimal Value – Long Form Appendage Element

Long Form Decimal Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	4
Format Code	1	1	Numeric	2 – Long Form Decimal
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	4	Decimal	Implied 4 places of precision

2.1.1.1

Decimal Value – Extended Form Appendage Element

Extended Form Decimal Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	9
Format Code	1	1	Numeric	3 – Extended Form Decimal
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value Denominator Code	3	1	Numeric	Number of decimal places in Value Numerator
Value Numerator	4	8	Numeric	

2.1.1.1 Numeric Value – Byte Appendage Element

Byte Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	1
Format Code	1	1	Numeric	7 – Byte Value
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	1	Numeric	

2.1.1.1 Numeric Value – Short Appendage Element

Short Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	2
Format Code	1	1	Numeric	8 – Short Value

2.1.1.1

Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	2	Numeric	

Numeric Value – Int32 Appendage Element

Int32 Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	4
Format Code	1	1	Numeric	9 – Int32 Value
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	4	Numeric	

2.1.1.1 Numeric Value – Int64 Appendage Element

Int64 Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	8
Format Code	1	1	Numeric	10 – Int64 Value
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	8	Numeric	

2.1.1.1 String Appendage Element

String Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**

2.1.1.1

Element Length	0	1	Numeric	Length of the String contained in the Value field
Format Code	1	1	Numeric	15 – String
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	Variable	Alpha	Length of this field is contained in the Element Length field

Date Appendage Element

Date Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	4
Format Code	1	1	Numeric	16 – Date
Value Type Code	2	1	Numeric	See Value Types for specific messages
Month	3	1	Numeric	1-12
Day	4	1	Numeric	1-31
Year	5	2	Numeric	

Char Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	1
Format Code	1	1	Numeric	22 – Char Value
Value Type Code	2	1	Numeric	See Value Types for specific messages

2.1.1.1

Value	3	2	Character	

Boolean Appendage Element

Boolean Value Appendage Element				
Field Name	**Offset**	**Length**	**Type**	**Notes**
Element Length	0	1	Numeric	1
Format Code	1	1	Numeric	21 – Char Value
Value Type Code	2	1	Numeric	See Value Types for specific messages
Value	3	1	Numeric	0 = False, 1 = True

Char Appendage Element

2.2 System and Channel Messages

2.2.1 System Event

The System Event message is used to signal events associated with the operations of the Smart Options System.

System Event			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	0x20 – System Event
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Timestamp	nanos	Long	SMART OPTIONS System Timestamp, in Nanoseconds since Midnight
Event Code	event	String	See Event Codes in Appendix

2.2.1 Channel Seconds

The Channel Seconds message is used to update the seconds portion of the timestamp for the specified market data source protocol and channel index. **NOTE:** This message *only* updates the timestamp information for the specified Channel Index on the specified Protocol ID. The client *MUST* key this data into both fields!

Channel Seconds			
Field	**Name**	**Type**	**Notes**
Message Type	msgTyoe	String	0x22 – Channel Seconds
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.

Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	String	Source Channel Index
Seconds	seconds	Long	Seconds Since Midnight for this Protocol ID and Channel Index

2.3 Locate Messages

Locate Messages are used for different elements within the data feeds which tend to require a large amount of space per message to repeat within each message. Examples include Instrument Symbols, Market Center codes, and Market Maker IDs. L*ocate codes are unique only within the context of the locate type.* Locate codes are not carried over between days and are **not** unique across multiple channels.

2.3.1 Market Center Locate

Market Center Locates are unique amongst Market Centers only.

Market Center Locate			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	0x30 – Market Center Locate
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Locate Code	locate	Long	Market Center Locate Code
MIC	mic	string	ISO 10383 Market Identifier Code

2.3.2 Instrumental Symbol Locate

Instrument Locate messages are unique amongst all instrument types. The **Product Type** code defines the type of instrument. Optional Appendages may be included for Instruments with individual component information, such as Option Strike Prices and Expiration Dates. The Currency Code, Country Code, and MIC fields are included where they are necessary to uniquely identify an instrument; based on the product type. These fields will be space-filled for instruments where they are not required.

Instrument Symbol Locate			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	0x33 – Instrument Locate
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Instrument Symbol Locate			
Field	**Name**	**Type**	**Notes**
Locate Code	locate	Int	Instrument Locate Code
Country Code	country	String	ISO 3166 Country Code (may be blank)
Currency Code	currency	String	ISO 4217 Currency Code (may be blank)
MIC	mic	String	ISO 10383 Market Identifier Code (may be blank)
Product Type	productType	Int	See Product Type Table in Appendix
Symbol Length	symbolLen	Int	Length of Symbol Field (SL)
Symbol	symbol	Byte[]	SMART OPTIONS Symbol for this instrument (Length of this string is defined by Symbol Length field)
Appendage	appendages	Byte[]	

2.4 Market Data Messages

Message Flags are used to denote special information regarding a particular message. Nasdaq Supports the following Message Flags:

Message Flags	
Bitmask Value	**Explanation**
0x01	Message is being disseminated out of sequence due to retransmissions.

0x02	Message contains data generated as a result of recovery mechanisms; example, recovery snapshots. Data contained in this message may not be a one-to-one replica of original upstream events.
0x04	Message Timestamp contains the Smart Options System time the message was processed; not the upstream data feed time. This is used when the upstream data feed does not provide timestamp information on an update.

2.4.1 NBBO Messages

NBBO messages come in multiple formats – short, long, extended, along with two-sided and one sided. NBBO messages currently do not support any appendage value types; though this may change in future revisions, so the appendage lists have been included in the message layouts.

2.4.1.1 Two Sided NBBO Update – Short Form

Two Sided NBBO – Short Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x60 – Short 2-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	

Bid Market Center Locate	bidMCLocate	Int	
Bid Price	bidPrice	Long	Implied 2 decimal places
Bid Size	bideSize	Long	
Ask Market Center Locate	askMCLocate	Int	
Ask Price	askPrice	Long	Implied 2 decimal places
Ask Size	askSize	Long	
Condition	condition	Int	See NBBO Condition Table in Appendix
Flags	quoteFlags	Byte[]	See NBBO Flags Table in Appendix
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.1.1 Two Sided NBBO Update –Long Form

Two Sided NBBO – Long Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x61 – Long 2-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Instrument Locate	sLocate	Int	
Bid Market Center Locate	bidMCLocate	Int	
Bid Price	bidDenom	Int	Implied 4 decimal places
Bid Size	bidPrice	Long	
Ask Market Center Locate	bideSize	Long	
Ask Price	askPrice	Long	Implied 4 decimal places

Ask Size	askSize	Long	
Condition	condition	Int	See NBBO Condition Table in Appendix
Flags	quoteFlags	Byte[]	See NBBO Flags Table in Appendix
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.1.1 Two Sided NBBO Update – Extended Form

Two Sided NBBO – Extended Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x62 – Extended 2-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Bid Market Center Locate	bidMCLocate	Int	
Bid Price Denominator Code	bidDenom	Int	Number of decimal places in Bid Price
Bid Price	bidPrice	Long	
Bid Size	bideSize	Long	

Ask Market Center Locate	askMCLocate	Int	
Ask Price Denominator Code	askDenom	Int	Number of decimal places in Ask Price
Ask Price	askPrice	Long	
Ask Size	askSize	Long	
Two Sided NBBO – Extended Form			
Field	**Name**	**Type**	**Notes**
Condition	condition	Int	See NBBO Condition Table in Appendix
Flags	quoteFlags	Byte[]	See NBBO Flags Table in Appendix
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.1.1 One Sided NBBO Update – Short Form

One Sided NBBO – Short Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x63 – Short 1-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	

Market Center Locate	mcLocate	Int	
Price	price	Long	Implied 2 decimal places
Size	size	Long	
Side	side	String	B = Buy, S = Sell
Condition	condition	Int	
Flags	quoteFlags	Byte[]	See NBBO Flags Tables in Appendix
One Sided NBBO – Short Form			
Field	**Name**	**Type**	**Notes**
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.1.1 One Sided NBBO Update – Long Form

One Sided NBBO – Long Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x64 – Long 1-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	

Price	price	Long	Implied 4 decimal places
Size	size	Long	
Side	side	String	B = Buy, S = Sell
Condition	condition	Int	See NBBO Condition Table in Appendix
Flags	quoteFlags	Byte[]	See NBBO Flags Tables in Appendix
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.1.1 One Sided NBBO Update – Extended Form

Short Form Decimal Appendage Element			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x65 – Extended 1-Sided NBBO
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	
Price Denominator Code	denom	Int	Number of decimal places in Price
Price	price	Long	
Size	size	Long	
Side	side	String	B = Buy, S = Sell

Condition	condition	Int	See NBBO Condition Table in Appendix
Flags	quoteFlags	Byte[]	See NBBO Flags Tables in Appendix
Appendage List	appendages	Byte[]	None Supported at Current Time

2.4.2 Trade Messages

2.4.2.1 Trade – Short Form

Trade – Short Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x70 – Short Trade
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	
Trade ID	tradeId	Int	
Price	price	Long	Implied 2 decimal places
Size	volume	Long	
Price Flags	priceFlags	Byte[]	See Trade Price Flags Table in Appendix
Eligibility Flags	eligFlags	Byte[]	See Eligibility Flags Table in Appendix
Report Flags	reportFlags	Byte[]	See Report Flags Table in Appendix

Change Flags	cancelFlags	Byte[]	See Change Flags Table in Appendix
Appendage List	appendages	Byte[]	

2.4.2.1 Trade – Long Form

Trade – Long Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x71 – Long Trade
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	
Trade ID	tradeId	Int	
Price	price	Long	Implied 4 decimal places
Size	volume	Long	
Price Flags	priceFlags	Byte[]	See Trade Price Flags Table in Appendix
Eligibility Flags	eligFlags	Byte[]	See Eligibility Flags Table in Appendix
Report Flags	reportFlags	Byte[]	See Report Flags Table in Appendix
Change Flags	cancelFlags	Byte[]	See Change Flags Table in Appendix

Appendage List	appendages	Byte[]	

2.4.2.1 Trade – Extended Form

Trade – Extended Form			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x72 – Extended Trade
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	
Trade ID	tradeId	Int	
Price Denominator Code	denom	Int	Number of decimal places in Price
Price	price	Long	
Size	volume	Long	
Price Flags	priceFlags	Byte[]	See Trade Price Flags Table in Appendix
Eligibility Flags	eligFlags	Byte[]	See Eligibility Flags Table in Appendix
Report Flags	reportFlags	Byte[]	See Report Flags Table in Appendix
Change Flags	cancelFlags	Byte[]	See Change Flags Table in Appendix

Appendage List	appendages	Byte[]	All values being updated are contained in appendage list

2.4.2.1 Trade – Extended Form

Trade Cancels *may* contain details on the original trade if the upstream data feed provides it. Some feeds refer to the corrected trade only by Original Trade ID, while others do not carry a Trade ID and simply provide all of the original details. "**Original**" fields will be populated if data is available. If data is unavailable, these fields will contain a value of 0. The CancelFlags will also be used to denote if the original trade information was unavailable from the source (see Appendix).

Trade Cancel			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x73 – Trade Cancel
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Market Center Locate	mcLocate	Int	
Original Trade ID	tradeId	Int	0 if Not Applicable
Original Price Denominator	denom	Int	Number of decimal places in Original Price
Original Price	price	Long	0 if Not Applicable

Original Size	volume	Long	0 if Not Applicable
Original Price Flags	priceFlags	Byte[]	See Trade Price Flags Table in Appendix
Trade Cancel			
Field	**Name**	**Type**	**Notes**
Original Eligibility Flags	eligFlags	Byte[]	See Eligibility Flags Table in Appendix
Original Report Flags	reportFlags	Byte[]	See Report Flags Table in Appendix
Cancel Flags	cancelFlags	Byte[]	See Trade Cancel Flags in Appendix
Appendage List	appendages	Byte[]	

2.4.3 Value Updates

Value Update message(s) is the value information that is not part of a quote that is disseminated on Smart Options. Examples of this type of information include common summary values, such as the High and Low Prices of the day, Index Tick values, Trading Range Indications, and other similar data.

2.4.3.1 Instrument Value Update

Value Update			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x80 – Value Update
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Upstream Sequence Number	srcSequence	Int	

Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	
Value Update			
Field	**Name**	**Type**	**Notes**
Market Center Locate	mcLocate	Int	0 if Not Applicable
Value Update Flags	flags	Int	See Value Update Flags in Appendix
Appendage List	appendages	Byte[]	All values being updated are contained in appendage list

2.4.4 Symbol State Messages

2.4.4.1 Instrument Status

The **Instrument Status** message is used to convey state changes for a given instrument, such – as Trading Halts and Reg-SHO events – which affect the market as a whole. The **Status Type** field determines what kind of status update is being conveyed (Trading State, Reg-SHO, etc.), the **Status Code** denotes the new value for that status type, the **Reason Code** conveys a normalized reason code (if a reason is available for the change), and the **Reason Detail** conveys the upstream reason details (if available).

Instrument Status			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0x90 –Trading Action
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below

Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	locate	Int	
Instrument Status			
Field	**Name**	**Type**	**Notes**
Market Center Locate	mcLocate	Int	
Status Type	statusType	Int	See Status Types in Appendix
Status Code	statusCode	Int	See Status Codes In Appendix
Reason Code	reason	Int	Reserved for future use
Status Flags	statusFlags	Byte[]	Reserved for future use
Reason Detail Length	detailLen	Int	Length of Reason Detail Field
Reason Detail	detail	Byte[]	Reason Detail String (If Available) – Length of this field is determined by the Reason Detail Length field
Appendage List	appendages	Byte[]	

2.4.5 Other Market Data Messages

2.4.5.1 Channel Event

The **Channel Event** message is used to convey market events published on a specific upstream channel and (optionally) for a specific Market Center. An optional appendage list is also provided for extra details on the event; if present in the upstream data feed. For example, some feeds will fire an event for a specific group of instruments. If this is the case, the group will show up as an appendage to this event and also as meta-data for the instrument.

Channel Event			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0xB0 – Channel Event

SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below
Channel Event			
Field	**Name**	**Type**	**Notes**
Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Instrument Locate	sLocate	Int	See Channel Event Codes below
Market Center Locate	mcLocate	Int	0 if Not Applicable to a specific Market Center
Appendage List	appendages	Byte[]	

2.4.5.1 Administrative Text

Administrative Text			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	MsgType 0xB2 – Admin Text
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Protocol ID	protocol	String	Source Protocol ID
Channel Index	channelIndex	string	Source Channel Index
Message Flags	msgFlags	String	See Message Flags Table below

Upstream Sequence Number	srcSequence	Int	
Upstream Nanos	srcNanos	Int	Nanosecond portion of the Upstream Timestamp for this Protocol ID and Channel Index (see Channel Seconds Message)
Text Length	textLen	Int	Length of Text Field
Text	text	Byte[]	Text Message – Length of this field is determined by the Text Length field

2.5 Miscellaneous and Reference Data Messages

2.5.1 Option Delivery Component

The Option Delivery Component message is used to describe a delivery component of a non-standard option. These components apply to the Root Code, rather than a specific contract.

Option Underlying			
Field	**Name**	**Type**	**Notes**
Message Type	msgType	String	0xC3 – Option Delivery Component
SOUP Partition	SoupPartition	Int	Message partition identifier. Ignore.
SOUP Sequence	SoupSequence	Long	Auto-incrementing message sequence number.
Root Code Locate	rLocate	Int	Symbol Locate for Option Root Code
Component Index	index	Int	Index of this component within the total component count for this Root Code
Component Total	total	Int	Total Expected Number of delivery components for this root. A value of zero indicates no special components available
Deliverable Units	deliverableUnits	Int	
Settlement Method	settlementMethod	String	Settlement Method 1 = Cash 2 = Continuous Net Settlement 3 = Broker to Broker

Fixed Amount Denominator	fixedAmtDenom	String	Number of decimal places in Fixed Amount Numerator
Fixed Amount Numerator	fixedAmt	Long	Fixed Amount – used for Cash Settlement
Currency Code	currency	String	ISO 4217 Currency Code (may be blank if not Cash Settlement – refer to Component Symbol Locate field)
Strike Percent	strikePct	int	Percentage of Strike Price allocated for Settlement - Implied 2 decimal places
Component Symbol Locate	cLocate	int	Symbol Locate for Component Symbol (0 if Cash Settlement – refer to Currency Code field)

Appendix A: Enumerations and Bitmask Values

This appendix contains all of the enumerations and bitmask values used in the Nasdaq Smart Options protocol. Bitmasks are represented in hexadecimal. Enumerations will utilize single quotes to denote an ASCII character value from a numeric value.

A.1 Message Flags

Market Data messages from the Smart Options feed contain the MsgFlags bitmask field, which provides information on the message itself; instructing the client on potential special handling that may need to be done with the message.

MsgFlags – BitMasks	
Code	**Explanation**
0x01	Out of Sequence – Message is being provided out of sequence – client should inspect the upstream sequence number provided *before* applying the update.
0x02	Reserved for future use.
0x04	Internal Timestamp – The original upstream message did not provide a timestamp – the SMART OPTIONS system has provided the SMART OPTIONS system timestamp for which the upstream message was received instead.

A.2 Appendage Value Types

Each of the Appendage Elements described in the Smart Options Specification contain a byte for the Value Type Code. The Value Type Codes are described in the table below, "Appendage Value Types". If the Value Type is 255, it means that the contents of the value are another Appendage Element (containing a length, a format code, a Value Type Code, etc.), for which Value Type is described in section 3.2, "Extended Value Types", below. **Note that in the event further extensions are required, they will be done in a recursive fashion**: Extended Value Type 255 will indicate that the value contains an appendage with Value Types taken from a table of Extended-Extended Value Types, and so on.

Byte Enumerations and Bitmasks are appended using the Byte Numeric Value format type. Each enumeration's values are described later in this document. **Note: Value Types may not be available from the upstream data sources for different instruments.** Smart Options will provide them via Value Updates, Locate messages, or Meta-Data messages *only* if provided via one of the upstream data sources. If the data is not provided, no appendage will be generated for the value type.

Appendage Value Types				
Value Type Code	**Value**	**Format**	**Message Types**	**From OPRA Message(s)**
1	Root Symbol	String	Instrument Locate	
2	Put/Call	Char	Instrument Locate	
3	Expiration Date	Date	Instrument Locate	
4	Strike Price	Decimal Value	Instrument Locate	
38	Strategy	Byte Enum	Trades, Trade Cancels	Last Sale
65	High Price	Decimal Value	Value Update	EOD Summary
66	Low Price	Decimal Value	Value Update	EOD Summary
67	Last Price	Decimal Value	Value Update	EOD Summary
68	Open Price	Decimal Value	Value Update	EOD Summary
70	Total Volume	Numeric Value	Value Update	EOD Summary
71	Net Change	Decimal Value	Value Update	EOD Summary

72	Open Interest	Numeric Value	Value Update	Open Interest EOD Summary
73	Tick	Decimal Value	Value Update	Underlying Last Sale
74	Bid	Decimal Value	Value Update	EOD Summary
75	Ask	Decimal Value	Value Update	EOD Summary
100	Underlying Price	Decimal Value	Value Update	EOD Summary
143	Upstream Condition Details	String	Trades, Trade Cancel	Last Sale
145	Trade Report Detail	Byte Enum	Trades, Trade Cancel	Last Sale
146	Extended Report Flags	Numeric Value	Trades, Trades Cancel	Last Sale

A.3 Cancel Flags

The Cancel Flags are represented in a bitmask. These are sent to denote “special” flags on a trade cancellation.

Cancel Flags Bits	
Code	**Explanation**
0x01	Opening (cancel was for the opening trade)
0x02	Error (cancel was denoted as an Error)
0x04	Original Trade Details Not Provided

A.4 Change Flags

The Change Flags are represented in a bitmask. These are sent on trade-related events to denote that the event has updated a specific statistic field for the instrument. These flags are used *only* for statistical fields that are updated on a large percentage of the trades. Statistical field modifications that do not occur as often as the fields denoted in these flags, such as the High and Low prices, are sent in a Value Update message following the trade event if they are updated.

Change Flags Bits

Code	Explanation
0x01	Consolidated Volume Updated (apply the volume of the trade event)
0x02	Market Center Volume Updated (apply the volume of the trade event)
0x04	Consolidated Last Sale Price Updated (apply the price of the trade event)
0x08	Market Center Last Sale Price Updated (apply the price of the trade event)

A.5 Channel Events

Channel Event Codes	
Code	**Explanation**
*SOD	Start of Day
*EOD	End of Day
Channel Event Codes	
*EOT	End of Transactions
*SOI	Start Open Interest
*EOI	End Open Interest
*SOS	Start of Summaries
*EOS	End of Summaries
*GDM	Good Morning
*GDN	Good Night

A.6 Eligibility Flags

The Eligibility Flags are represented in a bitmask and sent with Trade/Cancel/Correction events to denote what statistical values a trade is eligible to update. Note that this does not mean the trade always updates that statistic. Rather, it denotes only that it is *eligible* to do so. For example, a trade can be eligible to update the consolidated high price, but if it does not have a price higher than the existing high price, it will not do so.

Eligibility Flags Bits	
Code	**Explanation**
0x01	Consolidated Last Price
0x02	Consolidated Volume
0x04	Consolidated High/Low
0x08	Market Center Last Price
0x10	Market Center Volume
0x20	Market Center High/ Low
0x40	Market Center Open Price
0x80	Market Center Close Price

A.7 NBBO Flags

NBBO Flags Bitmask	
Code	**Explanation**
0x01	Quote Sizes are Provided in Shares, not Lots

A.8 Price Flags

The Price Flags are represented in a bitmask – any number of the following flags may be present in a price flags field. This field is currently always set to 0 but may be used in the future.

A.9 Product Types

The product type code is sent in the Instrument Locate message.

Product Type Codes	
Code	**Explanation**
0	Unknown
1	Equity
2	Option

3	Index
23	Option Root

A.10 Protocol IDs

Protocol IDs	
Code	**Explanation**
1 through 50	Reserved for future use
51	OPRA v3 (Binary)
52 through 255	Reserved for future use

A.11 Put/Call

NBBO Condition Values	
Code	**Explanation**
'P'	Put
NBBO Condition Values	
'C'	Call

A.12 Trade Report Flags

The **Trade Report Flags** are represented in a bitmask. These are sent to denote the most commonly disseminated or filtered flags on trade events. NOTE: For certain "special" reporting events on US Equities feeds that denote a Market Center's Official Opening or Official Closing price, but do not represent an actual trade (i.e. are not eligible to update the volume counters), the "Official Price Report" bit will be set along with either the "Opening" bit (to denote an official opening price for that market center), or the "Closing" bit (to denote an official closing price for that market center). These reports will not be flagged as volume eligible. It is important to note that the "Opening" and "Closing" bits may also be set on valid reports to denote an Opening or Closing print. These reports will be marked as Volume Eligible, and will not have the "Official Price Report" bit set.

Trade Report Flags Bits	
Code	**Explanation**

0x0001	Opening
0x0002	Closing
0x0004	Re-Opening
0x0008	Cross Trade
0x0010	Extended Hours
0x0020	Intermarket Sweep (ISO)
0x0040	Trade Through Exempt (US Equities)
0x0080	Odd Lot (Equities)
0x0100	Official Price Report (Used in conjunction with Opening and Closing flags)
0x0200	Floor (Futures)
0x0400	Summary (Used to flag auction summary)
0x0800	Printable

A.13 Trade Report Extended Flags

These flags are sent as a bitmask appendage to Trade/Cancel/Correction events to denote rare non-regular report information. This appendage will **not** be sent if there aren't any flags set. This appendage will be sent for Correction events for the corrected trade details only.

Trade Report Extended Flags Bits	
Code	**Explanation**
0x0001	Out of Sequence
0x0002	Late Reporting
0x0020	Stopped

A.14 Status Changes

SMART OPTIONS sends two types of Status Change events – **Instrument Status Change** events are market wide events and **Market Center Actions** are events that are occurring on a single Market Center.

A.14.1 Status Type

The **Status Type** enum is sent on Instrument Status and Market Center Action messages to denote what type of status update is being provided.

Status Type Values	
Code	**Explanation**
1	Trading State

A.14.2 Status Code

The **Status Code** field is sent on Instrument Status and Market Center Action messages to denote the actual value of the status being updated. This has multiple tables, and the table used is dependent on the value of the Status Type field.

A.14.2.1 Status Code – Status Type "Trading State"

Status Code – Trading State - Values	
Code	**Explanation**
1	Trading
2	Reserved for future use
Status Code – Trading State - Values	
3	Halted (Orders are not accepted or executing)

A.15 Strategy

Strategy – Values	
Code	**Explanation**
0	Unknown
2	Benchmark
35	Straddle
60	Spread
61	Buy-Write
62	Combo

A.16 System Events

Nasdaq supports the following System Event codes on Smart Options:

System Event Codes	
Code	**Explanation**
*Nul	Start of Messages: This is the first message sent in any market data session
*EOM	End of Messages: This is the last message sent in any market data session

A.17 Trade Report Detail

The **Trade Report Detail** enumeration is used to convey "extra" information on a trade. For some Option trades, this could contain the complex strategy used; for some Equity trades, this can convey whether the report is a bulk print from an auction, etc.

Trade Report Detail Codes	
Code	**Explanation**
0	None or Not Provided
7	AutomaticExecution
12	Adjustment
Trade Report Detail Codes	
13	Stopped Stock (No Trade Through)

A.18 Value Update Flags

These flags are sent as a bitmask on Value Update messages to denote special information regarding the update; for example, type of update – whether it was a result of a summary update.

Value Update Flag Bits	
Code	**Explanation**
0x0001	Summary
0x0002	Reserved for future use
0x0004	Reserved for future use
0x0008	End of Day

0x0010	Reserved for future use