

Symbol Discovery and Security Definitions Messages

- [EXCHANGE_LIST_REQUEST](#)
- [EXCHANGE_LIST_RESPONSE](#)
- [SYMBOLS FOR EXCHANGE REQUEST](#)
- [UNDERLYING SYMBOLS FOR EXCHANGE REQUEST](#)
- [SYMBOLS FOR UNDERLYING REQUEST](#)
- [SYMBOL_SEARCH_REQUEST](#)
- [SECURITY_DEFINITION_FOR_SYMBOL_REQUEST](#)
- [SECURITY_DEFINITION_REJECT](#)
- [SECURITY_DEFINITION_RESPONSE](#)

EXCHANGE_LIST_REQUEST [s_ExchangeListRequest structure] Client >> Server

This is a message from the Client to the Server to request a list of all available exchanges from the Server.

The server will respond with a separate [EXCHANGE_LIST_RESPONSE](#) message for each exchange.

In the case where the Server does not specify an exchange with its symbols, then the Server should provide a single response with an empty Exchange.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.

[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the EXCHANGE_LIST_RESPONSE message.
---	--

EXCHANGE_LIST_RESPONSE [s_ExchangeListResponse structure] Server >> Client

The server will return this message for each supported exchange.

If there are no exchanges to return in response to a request, send through one of these messages with the RequestID set and IsFinalMessage = 1. Leave all other members in the default state and the Client will recognize there are no Exchanges.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The RequestID sent in the request by the Client.
[char] Exchange	The exchange identifier that the Server uses to identify a particular exchange.
[unsigned int8] IsFinalMessage	1 = final message in batch.
[char] Description	The complete exchange description.

SYMBOLS_FOR_EXCHANGE_REQUEST

[s_SymbolsForExchangeRequest structure] Client >> Server

This is a message from the Client to the Server to request all of the Symbols for a particular Exchange.

The server will return a [SECURITY_DEFINITION_RESPONSE](#) message to the Client for each Symbol returned.

If the Server is rejecting this request, then it needs to send a [SECURITY_DEFINITION_REJECT](#) message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the SECURITY_DEFINITION_RESPONSE message.
[char] Exchange	The Exchange to return the available symbols listed on that Exchange.
[SecurityTypeEnum] SecurityType	The optional Security Type.

UNDERLYING_SYMBOLS_FOR_EXCHANGE_REQUEST

[s_UnderlyingSymbolsForExchangeRequest structure] Client >> Server

This is a message from the Client to the Server to request all of the underlying symbols on a particular Exchange. For example, all of the underlying futures symbols on a particular Exchange.

The server will return a [SECURITY_DEFINITION_RESPONSE](#) message to the Client for each

Symbol returned.

If the Server is rejecting this request, then it needs to send a [SECURITY_DEFINITION_REJECT](#) message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the SECURITY_DEFINITION_RESPONSE message.
[char] Exchange	.
[SecurityTypeEnum] SecurityType	.

SYMBOLS_FOR_UNDERLYING_REQUEST

[s_SymbolsForUnderlyingRequest structure] Client >> Server

This is a message from the Client to the Server for requesting all of the symbols for a particular underlying symbol.

For example, all of the futures contracts for a particular underlying futures symbol or all of the option symbols for a specific futures or stock symbol.

The server will return a [SECURITY_DEFINITION_RESPONSE](#) message to the Client for each Symbol returned.

If the Server is rejecting this request, then it needs to send a [SECURITY_DEFINITION_REJECT](#) message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the SECURITY_DEFINITION_RESPONSE message.
[char] UnderlyingSymbol	The underlying symbol.
[char] Exchange	The exchange of the symbols to search for.
[SecurityTypeEnum] SecurityType	The security type of the symbols to search for.

SYMBOL_SEARCH_REQUEST [s_SymbolSearchRequest structure] **Client >> Server**

The **SYMBOL_SEARCH_REQUEST** message is sent by the Client to the Server to return Security Definitions matching the specified **SecurityType** and **Exchange** and where the Symbol or Description contains the specified **SearchText**.

The **SearchText** can search either the Symbol or the Description field in the [SECURITY_DEFINITION_RESPONSE](#) message.

In either case there does not need to be an exact match. The **SearchText** only needs to be contained within the Symbol or the Description depending upon which field is being searched.

The Server returns [SECURITY_DEFINITION_RESPONSE](#) messages for all Symbols which match.

If there are no matches, the Server needs to send a [SECURITY_DEFINITION_RESPONSE](#) message to the Client with all fields at their default values except for the **RequestID** and **IsFinalMessage** fields set. This will be a clear indication to the Client that the request returned no matches.

If the Server is rejecting this request, then it needs to send a [SECURITY_DEFINITION_REJECT](#) message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the SECURITY_DEFINITION_RESPONSE message.
[char] SearchText	<p>The search text to search the Symbol or the Description for.</p> <p>If the SearchText field is an empty text string, then the server should reject the request with a SECURITY_DEFINITION_REJECT message.</p>
[char] Exchange	The Exchange of the Symbol to search for.
[SecurityTypeEnum] SecurityType	The Security Type of the Symbol to search for.

[SearchTypeEnum] SearchType	This field is the search type. Can be one of two possible values. Can specify to search the Symbol or the Description.
---	--

SECURITY_DEFINITION_FOR_SYMBOL_REQUEST **[s_SecurityDefinitionForSymbolRequest structure] Client >> Server**

This is a message from the Client to the Server for requesting Security Definition data for a specific symbol.

The Server will return a single [SECURITY_DEFINITION_RESPONSE](#) message in response to this request.

The Client must always send a **SECURITY_DEFINITION_FOR_SYMBOL_REQUEST** message to the Server in order to obtain the **IntegerToFloatPriceDivisor** value in case the Server uses the integer market data messages.

If the Server is rejecting this request, then it needs to send a [SECURITY_DEFINITION_REJECT](#) message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The unique identifier for this request. This same identifier will be returned in the SECURITY_DEFINITION_RESPONSE message.
[char] Symbol	The symbol to return a security definition for.

[char] Exchange	The optional exchange for the Symbol.
---	---------------------------------------

SECURITY_DEFINITION_REJECT [s_SecurityDefinitionReject structure] Server >> Client

This is a message from the Server to the Client indicating the Server is rejecting one of the following messages: [SYMBOLS_FOR_EXCHANGE_REQUEST](#), [UNDERLYING_SYMBOLS_FOR_EXCHANGE_REQUEST](#), [SYMBOLS_FOR_UNDERLYING_REQUEST](#), [SECURITY_DEFINITION_FOR_SYMBOL_REQUEST](#), [SYMBOL_SEARCH_REQUEST](#).

If there are no symbols to send in response to one of these messages above, then the Server should not send a [SECURITY_DEFINITION_REJECT](#) message and instead send a [SECURITY_DEFINITION_RESPONSE](#) with only the RequestID and IsFinalMessage fields set. This will be a clear indication to the Client that the request returned no Symbols.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	This is the same RequestID that this message is in response to and was given in the original request message.
[char] RejectText	Text reason for rejection.

SECURITY_DEFINITION_RESPONSE [s_SecurityDefinitionResponse structure] Server >> Client

This is a response from the Server in response to a [SYMBOLS_FOR_EXCHANGE_REQUEST](#), [UNDERLYING_SYMBOLS_FOR_EXCHANGE_REQUEST](#), [SYMBOLS_FOR_UNDERLYING_REQUEST](#), [SECURITY_DEFINITION_FOR_SYMBOL_REQUEST](#), [SYMBOL_SEARCH_REQUEST](#)

message.

If there are no symbols to return in response to a request, the Server needs to send through one of these messages with the **RequestID** set to the same RequestID value that the request message set it to, and **IsFinalMessage** = 1. Leave all other member fields in the default state and the Client will recognize there are no symbols.

The Client must always send a [SECURITY DEFINITION FOR SYMBOL REQUEST](#) message to the Server to obtain the **IntegerToFloatPriceDivisor** and **FloatToIntPriceMultiplier** values in the Security Definition Response message when the Server uses the integer market data and order messages.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Auto constructor.
[unsigned int16] Type	The standard message type field. Auto constructor.
[int32] RequestID	This is the same RequestID that this response is a response to and was given in the original request message.
[char] Symbol	<p>This is the Symbol the Security Definition Response is for.</p> <p>When the Server responds with a SECURITY DEFINITION RESPONSE message and there are no security definitions to return for the Symbol, this field will be empty.</p> <p>This field should be empty when this message is in response to a UNDERLYING SYMBOLS FOR EXCHANGE message.</p>
[char] Exchange	This is the Exchange for the Symbol. This field is only used when the Symbol is not empty.

[SecurityTypeEnum] SecurityType	The Security Type for the symbol.
[char] Description	The text description for the Symbol.
[float] MinPriceIncrement	The minimum amount that prices can move in the Symbol and minimum amount that prices can move in the Symbol.
[PriceDisplayFormatEnum] PriceDisplayFormat	This field specifies the price formatting for the Symbol.
[float] CurrencyValuePerIncrement	This field is the currency value per MinPriceIncrement in the Symbol's currency.
[unsigned int8] IsFinalMessage	Set to an integer value of 1 to indicate the last message in a batch of Security Definition messages.
[float] FloatToIntPriceMultiplier	With the integer order entry messages August 2020, this field is no longer relevant.
[float] IntegerToFloatPriceDivisor	With the integer market data messages August 2020, this field is no longer relevant.

<p>[char] UnderlyingSymbol</p>	<p>This is the underlying symbol for the Symbol. If the Symbol has an underlying symbol.</p> <p>The Server must set SECURITY_DEFINITION_RESPONSE response UNDERLYING_SYMBOLS_FOR_EXCHANGE SYMBOLS_FOR_UNDERLYING_REQUEST</p> <p>This is typically used with futures. A future specific contract year and month will have a symbol equivalent to the Symbol with month characters.</p>
<p>[unsigned int8] UpdatesBidAskOnly</p>	<p>This is set to 1 when the Symbol provides MARKET_DATA_UPDATE_TRADE when there is market activity for the Symbol.</p> <p>Otherwise, when this is MARKET_DATA_UPDATE_TRADE is received after subscribing to market data trading activity.</p>
<p>[float] StrikePrice</p>	<p>The strike price when the Security Type is an option.</p>
<p>[PutCallEnum] PutOrCall</p>	<p>When the Security Type is an option, this is put or call.</p>
<p>[unsigned int32] ShortInterest</p>	<p>The short interest when the Security Type is an equity.</p>
<p>[t_DateTime4Byte] SecurityExpirationDate</p>	<p>The expiration date for the Symbol for symbols with an expiration date.</p>

[float] BuyRolloverInterest	The daily interest amount which is de position. This only applies for Forex tr quote currency of the symbol.
[float] SellRolloverInterest	The daily interest amount which is de position. This only applies for Forex tr quote currency of the symbol.
[float] EarningsPerShare	The earnings per share as a currenc Security Type is a stock.
[unsigned int32] SharesOutstanding	This is the number of shares outstanding
[float] IntToFloatQuantityDivisor	With the integer order entry and mark discontinued as of August 2020, this relevant.
[unsigned int8] HasMarketDepthData	When HasMarketDepthData is set to Symbol has market depth data available set to 0, market depth data is not support

<p>[float] DisplayPriceMultiplier</p>	<p>This is an optional field for the Server to send to the Client.</p> <p>The default for this is 1.0.</p> <p>This sets the multiplier to use in the case the Server should multiply the values in market data by some number other than 1.0 before displaying to the user.</p> <p>It is recommended that a Server does not multiply values but instead transmit to the Client values as point values.</p> <p>This should not be confused with the multiplier in the IntToFloatPriceDivisor messages and the IntToFloatPriceDivisor messages. DisplayPriceMultiplier is used when the Server transmits market data values in integer types and where those values may need to be scaled to the original value by this multiplier to get the correct value to display to the user.</p> <p>When this is set to a value other than 1.0, the MinPriceIncrement and the PriceDisplayIncrement are relative to the market data value. DisplayPriceMultiplier is applied.</p>
<p>[char] ExchangeSymbol</p>	<p>This is an optional field. This is the exchange symbol that corresponds with the Symbol field.</p> <p>This field should be empty when this message is received from the UNDERLYING SYMBOLS FOR EXCHANGE.</p>
<p>[t_DateTime4Byte] RolloverDate</p>	<p>This field applies to the Futures Security.</p> <p>This is the rollover date for the symbol. It is the typical time where trading transitions from one month to the next.</p>

[float] InitialMarginRequirement	<p>This field applies to the Futures Security</p> <p>This is the initial margin requirement a exchange, if available.</p>
[float] MaintenanceMarginRequirement	<p>This field applies to the Futures Security</p> <p>This is the maintenance margin requirem the exchange, if available.</p>
[char] Currency	<p>This is the currency that the Symbol tra in.</p>
[float] ContractSize	<p>In the case of when a Symbol is a variable indicates the size of the contract</p> <p>This is going to be an exchange specific :</p>
[unsigned int32] OpenInterest	<p>In the case of when a Symbol is a contra the number of outstanding contracts.</p>
[unsigned int8] IsDelayed	<p>This field will be 1 if the market data intentionally delayed by a certain Otherwise, this will be 0.</p>

*Last modified Friday, 28th May, 2021.