iLink Message Specification

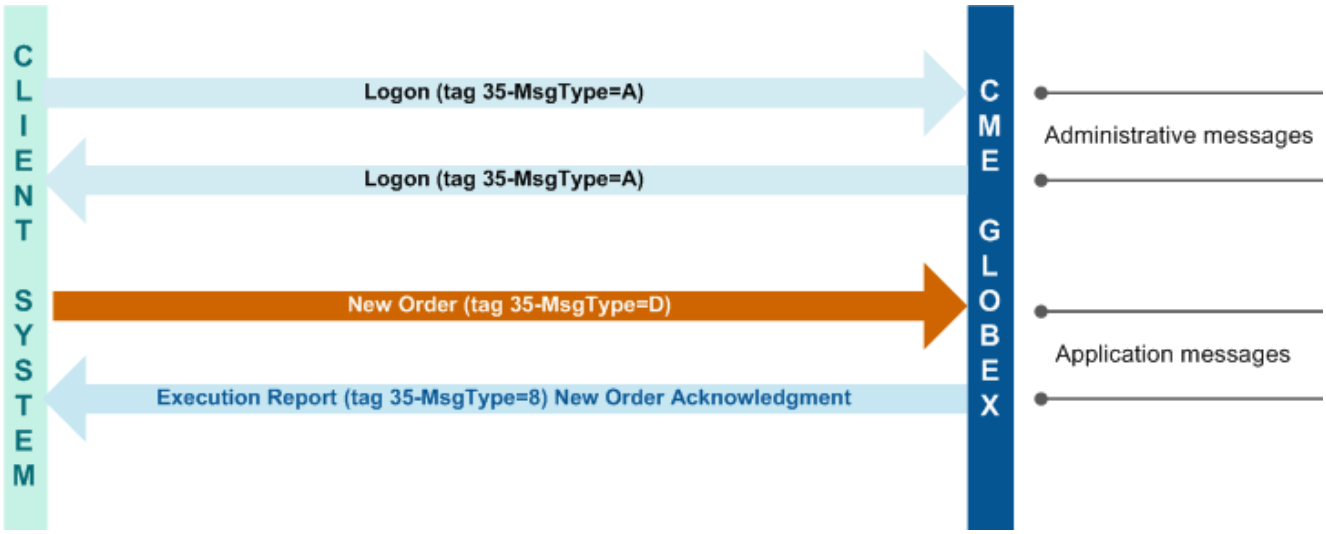
iLink is a bidirectional order entry path by which client systems connect to and transact trading activity on the CME Globex platform. For a trading session conducted on CME Globex over iLink, there are two layers of messaging, ***administration*** and ***application***.

- **Administration Messages** - support FIX session connectivity between the client system and CME Globex.
- **Application Messages** - support client system business transactions on CME Globex.

Use this search bar to search within the iLink message specification.

Both administration and application message types are sent between the client system and CME Globex throughout the trading session.

> For messages, iLink utilizes and accepts **printable** ASCII characters only.
>
> Extended ASCII codes and non-ASCII character sets are not permitted.



Using the iLink Message Specification

This section describes the features common to each message layout in the iLink message specification.

> To avoid client systems impacts, CME Group recommends that client systems **NOT** submit iLink messages containing FIX tags that are not defined in the iLink message specification.

Standard Header and Trailer

For each FIX4.2 message type sent over iLink, there is a standard header for outbound messages (from client system to CME Globex) and a standard header for inbound messages (from CME Globex to client system). The header for inbound and outbound FIX 4.2 message types sent over iLink is presented separately and referred to in subsequent message type layouts as 'Standard Header'. There is also a standard trailer for all FIX 4.2 message types.

Terminology

Each iLink message is comprised of message tags that are defined within the FIX 4.2 protocol. Each tag has its own set of attributes including:

Tag – tag number as assigned by the FIX 4.2 protocol

Name – tag name as assigned by the FIX 4.2 protocol

Req – indicates if tag must be sent on FIX 4.2 message

- Y=required by FIX 4.2 protocol
- Y*=required by iLink implementation
- N=tag is optional. If a non-required tag is sent by the Client System to CME Globex, it is returned on the inbound message as originally sent.
- C=tag is conditional. Refer to the description to determine applicability.

Valid Values – values accepted for this FIX tag for the given message type.

Repeating Group - it is permissible for fields to be repeated within a repeating group. Repeated instances are delimited by a tag, which is by definition the first field in the repeating group. If a repeating group is used, the first field of the repeating group is required. This allows implementations of the Protocol to use the first field as a "delimiter" indicating a new repeating group entry. If a repeating group field is listed as required, then it must appear in every repeated instance of that repeating group. Nested repeating groups are designated within the message definition with '' and a repeating tag within a repeating group is designated with double indentation ''.

Format

The format of each tag indicates the data type and maximum length for the tag value. Data types are defined by the FIX protocol as follows.

Data Type	Description
Int	Sequence of digits without commas or decimals and optional sign character (printable ASCII characters "-" and "0" - "9"). The sign character utilizes one byte (i.e. positive int is "99999" while negative int is "-99999"). Note that int values may contain leading zeros (e.g. "00023" = "23"). **Examples**: 723 in field 21 would be mapped int as 21=723 -723 in field 12 would be mapped int as 12=-723
Float	Sequence of digits with optional decimal point and sign character (printable ASCII characters "-", "0" - "9" and "."); the absence of the decimal point within the string will be interpreted as the float representation of an integer value. All float fields must accommodate up to fifteen significant digits. The number of decimal places used should be a factor of business/market needs and mutual agreement between counterparties. Note that float values may contain leading zeros (e.g. "00023.23" = "23.23") and may contain or omit trailing zeros after the decimal point (e.g. "23.0" = "23.0000" = "23" = "23."). Note that fields which are derived from float may contain negative values unless explicitly specified otherwise. The following data types are based on float. • **Qty** - float field capable of storing either a whole number (no decimal places) of "lots" (securities denominated in whole units) or a decimal value containing decimal places for non-share quantity asset classes (securities denominated in fractional units).Price - float field representing a price. Note the number of decimal places may vary. For certain asset classes prices may be negative values. For example, prices for options spreads can be negative under certain market conditions. • **PriceOffset** - float field representing a price offset, which can be mathematically added to a "Price". Note the number of decimal places may vary and some fields such as LastForwardPoints may be negative. • **Amt** - float field typically representing a Price times a Qty. • **Percentage** - float field representing a percentage (e.g. 0.05 represents 5% and 0.9525 represents 95.25%). Note the number of decimal places may vary.
Char	Single character value, can include any alphanumeric character or punctuation except the delimiter. All char fields are case sensitive. The following data type is based on Char. • **Boolean** - A character field containing one of two values: 'Y' = True/Yes, 'N' = False/No

String	Alpha-numeric free format strings, can include any character or punctuation except the delimiter. All char fields are case sensitive (i.e. morstatt ¹ Morstatt).
	• **MultipleStringValue** - String field containing one or more space delimited multiple character values.
	• **Currency** - String field representing a currency type.
	• **Exchange** - String field representing a market or exchange.
	• **UTCTimestamp** - Time/date combination represented in UTC (Universal Time Coordinated, also known as 'GMT') in either YYYYMMDD-HH:MM:SS (whole seconds) or YYYYMMDD-HH:MM:SS.sss (milliseconds) format. Colons, dash, and period are required. *Valid values:*
	  • YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-59 (without milliseconds).
	  • YYYY = 0000-9999, MM = 01-12, DD = 01-31, HH = 00-23, MM = 00-59, SS = 00-59, sss=000-999 (indicating milliseconds).
	Leap Seconds: Note that UTC includes corrections for leap seconds, which are inserted to account for slowing of the rotation of the earth. Leap second insertion is declared by the International Earth Rotation Service (IERS) and has, since 1972, only occurred on the night of Dec. 31 or Jun 30. The IERS considers March 31 and September 30 as secondary dates for leap second insertion, but has never utilized these dates. During a leap second insertion, a UTCTimestamp field may read "19981231-23:59:59", "19981231-23:59:59", "19990101-00:00:00". (see http://tycho.usno.navy.mil/leapsec.html)
	• **UTCTimeOnly** - Time-only represented in UTC (Universal Time Coordinated, also known as "GMT") in either HH:MM:SS (whole seconds) or HH:MM:SS.sss (milliseconds) format, colons, and period required. *Valid values*:
	  • HH = 00-23, MM = 00-59, SS = 00-59. (without milliseconds)
	  • HH = 00-23, MM = 00-59, SS = 00-59. sss=000-999 (indicating milliseconds).* LocalMktDate - Date of Local Market (vs. UTC) in YYYYMMDD format. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
	• **UTCDate** - Date represented in UTC (Universal Time Coordinated, also known as "GMT") in YYYYMMDD format. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.
	• MonthYear - char field representing month of a year in YYYYMM format. Valid values: YYYY = 0000-9999, MM = 01-12.

Data	String field containing raw data with no format or content restrictions. Data fields are always immediately preceded by a length field. The length field should specify the number of bytes of the value of the data field (up to but not including the terminating SOH). Caution: the value of one of these fields may contain the delimiter (SOH) character. Note that the value specified for this field should be followed by the delimiter (SOH) character as all fields are terminated with an "SOH".

See also:

- iLink FIX Tag Library
- Standard Header and Trailer Message Formats
- Administrative Message Formats
- Application Message Formats
- iLink Reject Codes
- Session Layer Validation Conditions

For more information regarding the FIX protocol, visit: http://www.fixprotocol.org/