



BMLL Data Feed

Level 2 quotes

Date: 2026-02-09

Version: 2.3

1.1. Introduction

The Level 2 quotes dataset offers a tick-by-tick view of the order book, aggregated by price level with up to 10 levels of depth. It provides an in-depth understanding of historical market activity, enabling users to make more informed trading decisions.

1.2. Dataset schema

Basic Types

Type	Content	Text representation
BOOLEAN	A True or False boolean value.	1,0
INT32	A 32 bit signed integer.	12345
INT64	A 64 bit signed integer.	12345
DOUBLE	IEEE 64-bit floating point value.	123.4567
BYTE_ARRAY	An arbitrarily long byte array.	"VODAFONE GROUP"
NUMBER	A number with defined precision (total digits allowed) and scale (number of digits allowed to the right of the decimal point). Unless otherwise specified, the default is precision 38 and scale 0.	12345
VARCHAR	Variable length string up to 500 characters. May contain quotes and special characters. All text will be utf8-encoded unless indicated differently. Ticker and ISO codes should be ASCII compatible.	"VODAFONE GROUP"
TIMESTAMP_NTZ	Nanoseconds since 1970 in UTC. Timestamp will be represented in ISO format in text format.	20220302-11:23.5712345789
FLOAT	IEEE 64-bit floating point value.	123.4567
DATE	A calendar date consisting of a year, month, and day, without including any time or time zone components.	2025-01-01

1.3. Schema

One row in the dataset corresponds to a single, atomic event during the trading day. Where the exchange explicitly provides events, only the final states of the event will be provided.

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
TICKER	Ticker	VARCHAR	BYTE_ARRAY	Ticker from FactSet
ISO_EXCHANGE_CODE	ISOExchangeCode	VARCHAR	BYTE_ARRAY	ISO Exchange Code from FactSet
TRADE_DATE	TradeDate	DATE	INT32	The specific date on which an order was placed - the date is derived from the LocalTimestamp.
EVENT_TIMESTAMP	EventTimestamp	TIMESTAMPNTZ	INT64	The UTC timestamp indicates the date and time the update occurred in the UTC timezone, with microsecond precision.
LOCAL_TIMESTAMP	LocalTimestamp	TIMESTAMPNTZ	INT64	The local timestamp indicates the date and time the update occurred in the timezone of the primary record associated with the venue, with microsecond precision.
TIMESTAMP_NANOSECONDS	TimestampNano seconds	NUMBER	INT64	This column represents the time of the update as the number of nanoseconds since 1970-01-01 00:00:00.0 UTC.
TZ_OFFSET	TZOffset	NUMBER	INT32	The offset between the local timestamp and UTC in seconds. This is based on the original listing location of the primary instrument. Format was +/- HHMM. This will be adjusted during the DST if such a change occurs.
Core MBP fields (Price levels 1-10 are available)				

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
BID_PRICE(1-10)	BidPrice[1-10]	DOUBLE	DOUBLE	The CBBO price at the given price level on the bid side in local currency.
BID_QUANTITY(1-10)	BidQuantity[1-10]	NUMBER	INT64	The number of shares available at the CBBO at the given price level on the bid side.
BID_NUM_ORDERS(1-10)	BidNumOrders[1-10]	NUMBER	INT32	The number of distinct orders at the CBBO at the given price level on the bid side.
ASK_PRICE(1-10)	AskPrice[1-10]	DOUBLE	DOUBLE	The CBBO price at the given price level on the ask side in local currency.
ASK_QUANTITY(1-10)	AskQuantity[1-10]	NUMBER	INT64	The number of shares available at the CBBO at the price level on the ask side.
ASK_NUM_ORDERS(1-10)	AskNumOrders[1-10]	NUMBER	INT32	The number of distinct orders at the CBBO at the given price level on the ask side.
Additional fields				
BID_LEVEL_COUNT	BidLevelCount	NUMBER	INT32	Number of bid levels populated.
ASK_LEVEL_COUNT	AskLevelCount	NUMBER	INT32	Number of ask levels populated.
L2_EVENT_NO	L2EventNo	NUMBER	INT64	L2EventNo enables the sorting of financial data that occur at the same timestamp. The first event of a particular day or specific event is labelled 1, with every subsequent update marked by an increase of 1.
MIC	MIC	VARCHAR	BYTE_ARRAY	The Market Identifier Code (MIC) uniquely identifies the exchange.
EXCHANGE_TICKER	ExchangeTicker	VARCHAR	BYTE_ARRAY	The original ticker as provided by the exchange.

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
BMLL_OBJECT_ID	BMLLObjectId	VARCHAR	INT64	The BMLL identifier representing the object.
CURRENCY_CODE	CurrencyCode	VARCHAR	BYTE_ARRAY	The currency of the order book information; as provided by the exchange.
BMLL_SEQUENCE_NO	BMLLSequenceNo	NUMBER	INT64	A unique identifier for each execution of a DAG.
BMLL_SEQUENCE_SOURCE	BMLLSequenceSource	NUMBER	INT64	Source of the internal sequence number. e.g. On-book or off-book trades, usually per input file.
EXCHANGE_SEQUENCE_NO	ExchangeSequenceNo	NUMBER	INT64	Sequence number from the exchange raw data.
MARKET_STATE	MarketState	VARCHAR	BYTE_ARRAY	Market state of the auction. See below for possible values.

1.4. Market State Normalisation

BMLL provides a consistent normalisation for market state messages, in order to make cross venue analysis quick and easy.

The following normalisation states are available

Market State

Value	Description
AUCTION_ON_DEMAND	Auctions which take place alongside CONTINUOUS_TRADING. Includes order entry and uncrossing periods.
CLOSED	The market is closed so no trading activity can take place.
CLOSING_AUCTION	Includes order entry and uncrossing periods.
CONDITIONAL	An uncrossing state for specific market mechanisms e.g. Turquoise Plato Uncross.
CONTINUOUS_TRADING	Main continuous trading session in which orders can be placed and matched. Separate continuous trading occurring before the primary open or after close will be generally indicated with PRE_TRADE and POST_TRADE.

CONTINUOUS_TRADING_PRIMARY_CLOSED	A continuous trading session in which orders can be placed and matched on secondary markets, but the primary market is CLOSED. Used for secondary markets only (e.g. a multilateral trading facility in the EU, or alternative trading system in the USA).
HALTED	Unscheduled trading halts, for example, when a circuit breaker is triggered.
INTRADAY_AUCTION	Scheduled auctions which interrupt CONTINUOUS_TRADING. Includes order entry and uncrossing periods.
NOT_APPLICABLE	Used for venues such as trade reporting facilities where the concept of market phase does not apply, as there are no specific market hours.
OPENING_AUCTION	Includes order entry and uncrossing periods.
POST_TRADE	Market phase following CONTINUOUS_TRADING phase, and generally occurring after the primary CLOSING_AUCTION. This includes trade-at-last phases; such trades can be distinguished using the CLOSING_PRICE bml_trade_type.
PRE_OPEN	Market phase preceding CONTINUOUS_TRADING, and generally prior to OPENING_AUCTION when applicable.
UNKNOWN	Used when the market phase could not be resolved into one of the known states listed here.
UNSCHEDULED_AUCTION	Unscheduled auctions which interrupt CONTINUOUS_TRADING. Includes order entry and uncrossing periods.

1.5. File format and delivery structure

Data is delivered as a single file per market per date, as a parquet file.

- **{YYYY}**: Year
- **{MM}**: Month
- **{DD}**: Date
- **{Product}**: The product you are subscribing
 - **Level 2 Quotes**: LEVEL2Q
- **{Region}**: The region of the venue such as APAC, AMERICAS, and EMEA
- **{MIC}**: Market Identifier Code (MIC) is a unique four-character code that identifies stock markets and exchanges for trading and referencing computer systems

File Structure:

{YYYY}/{MM}/{DD}/LEVEL2Q/{Region}/{MIC}-YYYYMMDD.parquet

1.6. Data arrival times and coverage

Full product coverage and arrival times are available at <https://data.bmlitech.com>.