BMLL Data Feed

Level 1 Quotes

Date: 2026-02-09
Version: 1.4

1.1. Introduction

A tick-by-tick view of the top level of the order book. Captures best bid and offer with sizes on every update, for every venue BMLL covers.
Use this dataset to study historical market activity, build quote-based signals, or reconstruct top-of-book conditions at any point in time.

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.	20220302-11:23.57.12345789
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

T: +44 (0)20 3828 9000
E: info@bmlltech.com
W: www.bmlltech.com
Copyright © 2026 BMLL Technologies. All rights reserved.

1.3. Schema

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
TICKER	Ticker	VARCHAR	BYTE_ARRAY	BMLL-format ticker for a listing on its trading venue. This is the customer-facing Ticker used in API, Snowflake, and Databricks materialisations.
ISO_EXCHANGE_CODE	ISOExchangeCode	VARCHAR	BYTE_ARRAY	The ISO code of the primary exchange associated with the Instrument. On stocks this ISO code will generally be one of, AMS, BRU, BUD, CSE, DUB, ETR, ICE, HEL, LIS, LIT, LON, MIL, MCE, OME, OSL, PAR, PRA, STO, SWX, RIS, TAL or WBO. In other classes, BATE, DXE, TQEX, TRQX may also be the main regional venue for the instrument.
TRADE_DATE	TradeDate	DATE	INT32	Date associated with the dataset row. Product-level smell records document any legacy product where this date is derived from publication time rather than the venue trade timestamp.
EVENT_TIMESTAMP	EventTimestamp	TIMESTAMP_NTZ	INT64	Time of the event - this is generally aimed to be matching event timestamp

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
LOCAL_TIMESTAMP	LocalTimestamp	TIMESTAMP_NTZ	INT64	The local timestamp indicates the date and time at which the update occurred in the timezone of the primary record associated with the venue.
TIMESTAMP_NANOSECONDS	TimestampNanoseconds	NUMBER	INT64	This column contains a representation of the time of the update as the number of nanoseconds since 1970-01-01 00:00:00.0 UTC.
TZ_OFFSET	TZOffset	NUMBER	INT64	The offset between the local timestamp and UTC in seconds.Note: 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.
BID_PRICE1	BidPrice1	FLOAT	DOUBLE	The CBBO price of the price level on the bid side in local currency
BID_QUANTITY1	BidQuantity1	NUMBER	INT64	The number of shares available in the EBBO at the price level on the bid side
BID_NUM_ORDERS1	BidNumOrders1	NUMBER	INT64	The number of distinct orders in the EBBO at given the price level on the bid side

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
ASK_PRICE1	AskPrice1	FLOAT	DOUBLE	The CBBO price of the price level on the ask side in local currency
ASK_QUANTITY1	AskQuantity1	NUMBER	INT64	The number of shares available in the nth most competitive price level on the ask side
ASK_NUM_ORDERS1	AskNumOrders1	NUMBER	INT64	The number of distinct orders in the EBBO at given the price level on the ask side
MIC	MIC	VARCHAR	BYTE_ARRAY	The Market Identifier Code (MIC) is used to uniquely identify the exchange which is reporting market activity.
EXCHANGE_TICKER	ExchangeTicker	VARCHAR	BYTE_ARRAY	BMLL-format venue ticker carried under the ExchangeTicker physical name in parquet/Revelate parquet schemas. API, Snowflake, and Databricks materialisations should expose this semantic as Ticker.

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
LISTING_ID	ListingId	NUMBER	INT64	The BMLL ListingId is used to identify a specific instrument on a market venue (trading venue or reporting venue). BMLL listings identify uniquely every ticker on the venue and remain attached to the instrument even when a ticker change occurs — though changes of primary instrument identifier (such as ISIN changes) can cause a new listing_id to be issued.
CURRENCY_CODE	CurrencyCode	VARCHAR	BYTE_ARRAY	The currency code used for all price information (unless specified differently in specific column descriptions). The Currency code is represented following ISO-4217 codes (see https://en.wikipedia.org/wiki/ISO_4217) . Minor currencies are represented using the major currency code and putting in lower case the last letter of the major currency code.
MARKET_STATE	MarketState	VARCHAR	BYTE_ARRAY	Standardised market state information. See below for description of possible market states.

Field Name (Snowflake / Databricks)	Field Name (Parquet)	Type (Snowflake / Databricks)	Type (Parquet)	Description
EXCHANGE_SEQUENCE_NO	ExchangeSequenceNo	NUMBER	INT64	The sequence number as provided by the exchange. Note that some exchange may not provide a sequence number in this case the ExchangeSequenceNo will always be 0.
BMLL_SEQUENCE_NO	BMLLSequenceNo	NUMBER	INT64	The sequence no of the trade in the upstream feed as inferred by BMLL - this sequence number will respect the order of messages in upstream protocol and will be populated even if the upstream venue does not provide a sequence number - where the source of the feed is indicated by BMLLSequenceSource.
BMLL_SEQUENCE_SOURCE	BMLLSequenceSource	NUMBER	INT64	An integer used to identify the upstream feed from which the sequence numbers are generated. This is designed to avoid the creation of non-existent order between events that come from different message streams

1.4. Market states

BMLL normalises venue-specific market-phase messages into a consistent set of states so cross-venue analysis is straightforward.

Canonical MarketState enumeration.

Code	Description
AUCTION_ON_DEMAND	An auction that occurs concurrently to the main trading session
CLOSED	The exchange is closed
CLOSING_AUCTION	Market in Closing Auction
CONDITIONAL	The trade is the result of a conditional trading mechanism

1.5. Format and File 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 1 Quotes:** LEVEL1
- **{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}/LEVEL1/{Region}/l1-{MIC}-{YYYYMMDD}.parquet
Example:
2026/02/09/LEVEL1/EMEA/l1-XLON-20260209.parquet