Data Transmission and Validation under the Hybrid EURIBOR Methodology

EMMI Benchmarks Application and System Software (EBASS)

The European Money Markets Institute

Confidential
D0098A-2019

Document information

Document title	**Data Transmission and Validation under the Hybrid EURIBOR Methodology**	Status:	Final version
Business area	Benchmarks Operations		
EMMI reference	D0098A-2019		
Sensitivity	Public Document		
Documentation type	Policy		
Review cycle	Bi-annual		

Table of contents

1. Preface

This document describes the process—as well as reporting instructions—for contributing input data for the EURIBOR benchmark under the hybrid EURIBOR methodology. The hybrid EURIBOR methodology is described in the Benchmark Determination Methodology for EURIBOR ('BDM'), which is available on the EMMI website. Under the hybrid methodology, Panel Banks contribute two types of input data; transactions for Level 1 and 2 processing, and, when needed, a Level 3 submission rate.

The input data will be sent to and processed by the EMMI Benchmarks Application and System Software (referred to as "EBASS" or "EMMI System"). The transactions data will be transmitted using the web service protocol described in Section 6. For Level 3, submission data will be transmitted using two alternative methods: the web service protocol or a manual input through the EBASS Graphical User Interface ('GUI'). Although the reporting instructions detailed in Section 3 apply primarily to the web service protocol, the information transmitted for Level 3 submission data will be the same if transmitted through the GUI.

The document is structured in six different sections, covering different aspects of the contribution process.

Section 2 – General approach and definitions provides a general overview of the change and some key principles that have been followed throughout the design and implementation of the specifications.

Section 3 – Delivery Message: This section describes the format of submission level data and transactions level data (EURIBOR) for transmission from Panel Banks to EMMI. Validations that are crossed out refer to past validations that have been discontinued, but are nonetheless included here for consistency in the validation codes.

Section 4 – General Formats: This section describes the particular data formats that apply to the input data and input data files.

Section 5 – Data Validations: This section covers the data validations that are performed on the input data and input data files to confirm that they conform to the specifications described elsewhere in this document.

Section 6 – Web services: This section provides an overview of the web service data transmission protocol.

2. General approach and definitions

2.1. Transaction Data VS Submission Data

This document differentiates between two levels of data reporting:

- **Transaction Data**—defines records reported as **Level 1 and 2** transactions under the new hybrid **EURIBOR** methodology;

- **Submission Data**—defines records reported as either (i) **Quotes** under the legacy **EURIBOR** methodology, or (ii) **Level 3** under the new Hybrid **EURIBOR** Methodology.

2.2. Abbreviations and acronyms

The following terms (arranged alphabetically) have been used throughout this document:

- **Contribution Window** – The daily period during which the submission infrastructure is available to accept data from Panel Banks. The window will vary depending on the type of information being shared, e.g. submission or transaction data;
- **EBASS** – EMMI Benchmark Application and System Software (also referred to as the "EMMI System");
- **EMMI** – European Money Markets Institute, the administrator of the EURIBOR benchmark;
- **GUI** – Graphical User Interface – an easy-to-use human-to-computer interface for managing the daily benchmark operations; it includes manual input and monitoring of EURIBOR submission data as well as monitoring of EURIBOR transaction data by Panel Bank users;
- **MMSR** – Money Market Statistical Reporting. A transaction-by-transaction reporting framework put in place by the Eurosystem to support the implementation of Regulation (EU) No 1333/2014 of the European Central Bank (ECB/2014/48)[1]. EMMI has chosen to emulate this framework for the daily transmission of transaction data. The standards have been enhanced to allow for the transmission of submission data to support the hybrid EURIBOR methodology;
- **Panel Banks** – Banks located in the EU and EFTA countries with business in the euro money markets who contribute data to EMMI for the purposes of calculating the EURIBOR rates;
- **Reporting Agent** – Used interchangeably with Panel Bank. This term is generally used in context of the message structure and validations to keep similarity with the terms and notations used by the ECB for the MMSR instructions;
- **TARGET2** – TARGET stands for Trans-European Automated Real-time Gross settlement Express Transfer system. The Eurosystem maintains TARGET2, which is the second generation of TARGET. Throughout this document, references to "TARGET" should be read with respect to the Eurosystem's TARGET2 system.

2.3. Synergies with ISO 20022/MMSR

As far as the reporting of transactions is concerned, EMMI has decided to mirror the European Central Bank's (ECB) framework for the Money Market Statistical Reporting (MMSR), where feasible. This approach seeks to minimize the burden on Panel Banks who also participate in the MMSR program, by allowing them to submit transaction data to EMMI in a format similar to that required by the ECB.

While the standard used by EMMI relies heavily on the ISO standard 20022 and MMSR format, enhancements have been made to accommodate the context in which this standard is being

[1] Regulation (EU) No 1333/2014 of the European Central Bank (ECB/2014/48) amended by Regulation (EU) No 1599/2015 of the European Central Bank (ECB/2015/30) and subsequently amended by Regulation ECB/2018/33

used for EMMI. Going forward, this EMMI standard may evolve separately from the evolution of the MMSR standard.

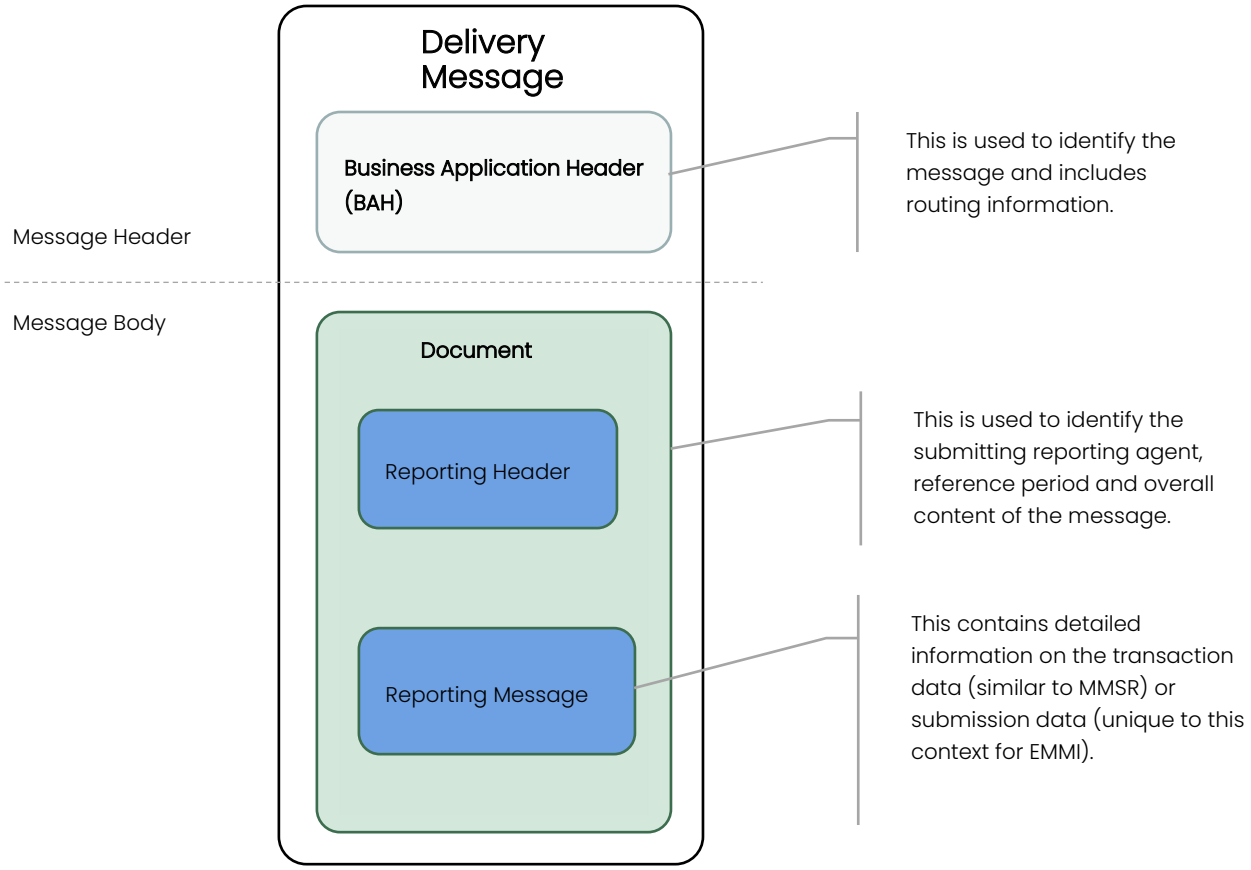
Furthermore, for reasons of consistency and synergies across benchmarks, EMMI has also chosen to leverage on the concepts and principles of ISO standard 20022 and MMSR for the reporting of submission data.

3. Delivery Message – Conceptual and Field Definitions

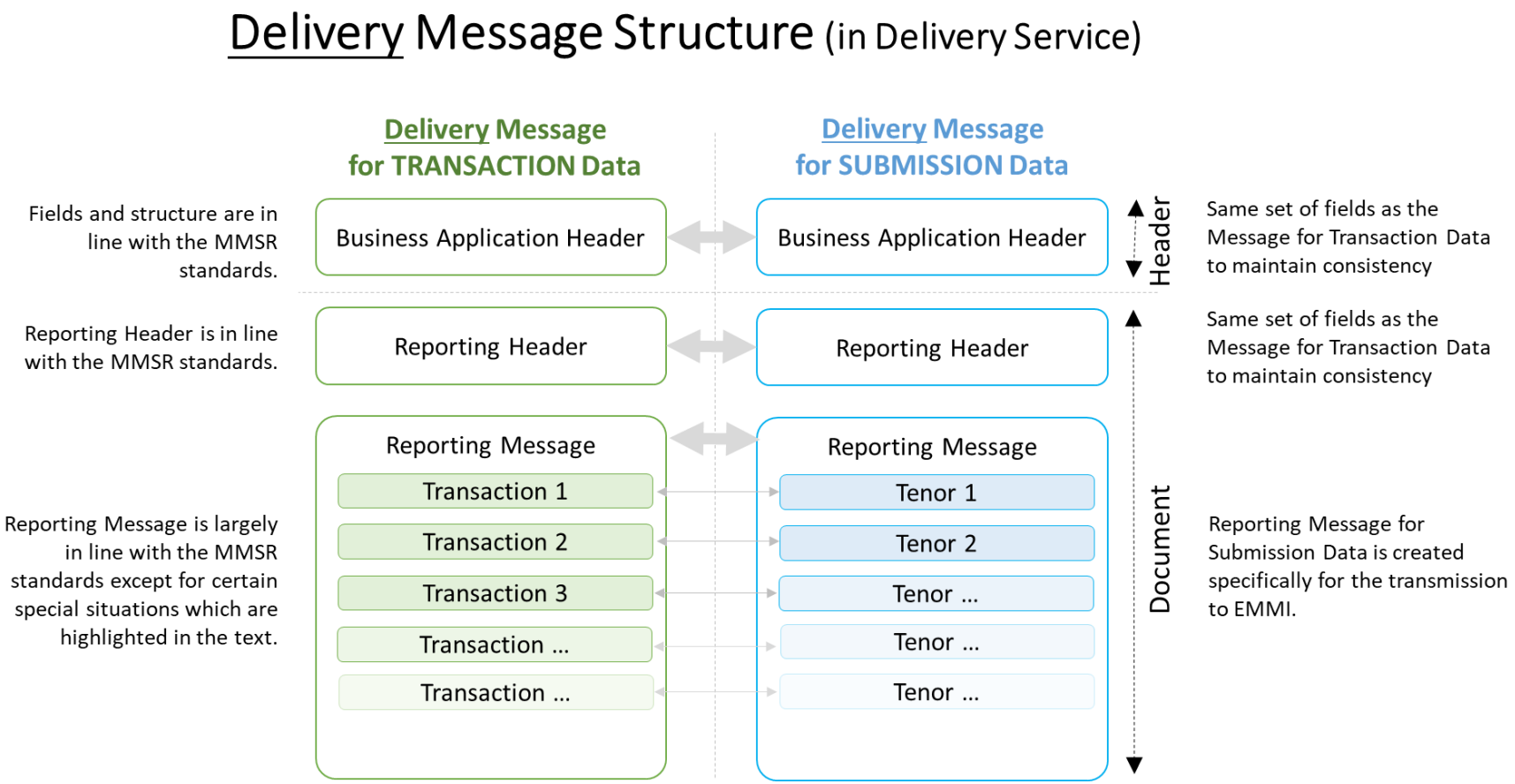
3.1. Conceptual Structure of a Delivery Message

In line with the MMSR framework, each EBASS delivery message is made up of the following structure:

- **Message Header:** This is used to identify the type of message and contains the routing information – who is this information coming from and who it is addressed to. The list of variables which compose the Message Header are described in section 3.2.
- **Message Body:** This is known as the "Document" and is made up of the following two parts:
  - **Reporting Header:** This is used to identify the reporting agent and the reference period. The list of variables which compose the Reporting Header are described in section 3.2.
  - **Reporting Message:** This contains details of the actual data that is being shared by the reporting agent. For transaction level data files, this section will contain the list of transactions and for submission level data file it will contain the list of submissions. Specific layout details of the individual Reporting Messages are described in section 0 (for submission data) and section 0 (for transaction data).


Delivery Message
Business Application Header (BAH)
Message Header
Message Body
Document
Reporting Header
Reporting Message
This is used to identify the message and includes routing information.
This is used to identify the submitting reporting agent, reference period and overall content of the message.
This contains detailed information on the transaction data (similar to MMSR) or submission data (unique to this context for EMMI).

The following diagram gives an overview of the difference between the delivery message structures for Transaction Data vs. Submission Data reporting:



3.2. Business Application Header

The following are the fields used in the Business Application Header (BAH) section for all benchmarks and data types:

3.2.1. Business Message Identifier

Variable ID	Variable Name	Type	Description	Example
H10	BUSINESS MESSAGE IDENTIFIER	String. Max length: 35 [A-Z][0-9]	A **character string** identifying the reporting agent followed by a non-repeating, alphanumeric string (A-Z, 0-9) for all files sent by the Panel Bank in order to uniquely refer to any given file in a bilateral communication. This Identifier must remain unique not only intraday but also from one day to the next. *In consultation with each Panel Bank, EMMI will determine an identifying character string for each Reporting Agent. Changes to these strings are possible and should be planned together with EMMI.*	**IREF**012345, or **BANKA**20181024ABC, or **BANKB**20181024092533

Category	Validation Identifier	Action	Description of validation	Timing
Error	~~EVH100~~	~~Reject File~~	~~Missing value for field BUSINESS MESSAGE IDENTIFIER.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH101	Reject File	Invalid format for field BUSINESS MESSAGE IDENTIFIER.	Synchronous
Error	EVH102	Reject File	AppHdr has the same BizMsgIdr and sender as a previous message.	Synchronous

3.2.2. Sender

Variable ID	Variable Name	Type	Description	Example
H20	SENDER	String. Length: 20 LEI	*Documents the sender of the message (either the reporting agent or another entity) using the Legal Entity Identifier (LEI). This variable is also named 'From' in the BAH.*	QS3ZEAHRBZY9228Z011

		[ISO17442]	*In consultation with each Panel bank, EMMI will configure the list of possible Senders for the Panel Bank and Reporting Agents that a Sender is allowed to report for. Should Panel Banks need to change this relationship, they are requested to contact EMMI to make suitable configuration changes before a new Sender can participate in the process.*	
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVH200~~	~~Reject File~~	~~Missing value for field SENDER.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH201	Reject File	Invalid LEI format [SENDER value] provided for field SENDER.	Synchronous
Error	EVH202	Reject File	Specified LEI [SENDER value] does not match the list of possible senders for this reporting agent on this segment.	Synchronous

3.2.3. Receiver

Variable ID	Variable Name	Type	Description	Example
H30	RECEIVER	String. Length: 20 LEI [ISO17442]	Documents the receiver of the message (EMMI) using the LEI. This variable is also named 'To' in the BAH.	BE0465075408EMMI0001
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVH300~~	~~Reject File~~	~~Missing value for field RECEIVER.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH301	Reject File	Invalid LEI format [RECEIVER value] provided for field RECEIVER.	Synchronous
Error	EVH302	Reject File	Specified LEI [RECEIVER value] does not match EMMI's LEI.	Synchronous

3.2.4. Business Service

Variable ID	Variable Name	Type	Description	Example
H40	BUSINESS SERVICE	String. Length: 9 (see Code List)	This variable specifies the service to which the receiver should route the reported data. The variable has two valid values: EMMI_PROD and EMMI_TEST. Code List: EMMI_PROD — This is used for normal benchmark contribution Code List: EMMI_TEST — This is used for connectivity testing purposes only **Messages assigned with EMMI_TEST will be issued a default "ACTC" synchronous status and will not be processed further.**	EMMI_PROD
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVH400~~	~~Reject File~~	~~Missing value for field BUSINESS SERVICE.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH401	Reject File	Invalid code for field BUSINESS SERVICE.	Synchronous

3.2.5. Market Segment Identifier

Variable ID	Variable Name	Type	Description	Example
H50	MARKET SEGMENT IDENTIFIER	String Length: 15 (see Code List)	This variable specifies the market segment of the subsequent reporting data in the message: transactions (unsecured market) or quotes. This variable is named 'Message Definition Identifier' in the BAH. Code List: auth.013.001.02 — Transaction data (unsecured segment of the money market) Code List: emmi.eurib.0001 — Submission data (EURIBOR quote)	auth.013.001.02 (for transaction data reporting)
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVH500~~	~~Reject File~~	~~Missing value for field MARKET SEGMENT IDENTIFIER.~~ *[This validation is a part of the XSD validations]*	Synchronous

Error	EVH501	Reject File	Invalid code for field MARKET SEGMENT IDENTIFIER. --- *Note: This is returned for all cases where the market segment identifier is not eligible for reporting to EMMI; even if it as a valid identifier as per MMSR standards (e.g. auth.012.001.02 for Secured segment).*	Synchronous
Error	EVH502	Reject File	Market Segment [MARKET SEGMENT IDENTIFIER value] is different from the Segment indicated in the Message File Name.	Synchronous
Error	EVH503	Reject File	Contribution Window for the Market Segment [MARKET SEGMENT IDENTIFIER value] is currently not open.	Synchronous

3.2.6. Creation date

Variable ID	Variable Name	Type	Description	Example
H60	CREATION DATE	Date-time YYYY-MM-DDThh:mm:ssZ	This is the date on which the file is generated. The format is as specified in ISO 20022, where it is aligned with ISO 8601.	2018-02-08T14:39:00Z

Category	Code	Action	Description of validation	Timing
~~Error~~	~~EVH600~~	~~Reject File~~	~~Missing value for field CREATION DATE.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH601	Reject File	Invalid date format [CREATION DATE value] detected for field CREATION DATE.	Synchronous
Error	EVH602	Reject File	Creation Date [CREATION DATE value] is before Reference Period <end> [Reference Period <end> value].	Synchronous
Error	EVH603	Reject File	Creation Date [CREATION DATE value] is in the future.	Synchronous

3.3. Reporting Header

The following are the fields used in the Reporting Header section for all benchmarks and data types:

3.3.1. Reporting Agent

Variable ID	Variable Name	Type	Description	Example
H70	REPORTING AGENT	String Max length: 20 [ISO17442]	This variable will contain the LEI of the reporting agent. In most cases, the Reporting Agent will be the same as the Sender (H20). However it may be the case that for a particular Panel Bank the sending entity is different from the reporting entity due for example to a specific organizational structure.	'QS3ZEAHRBZY9228Z0111' refers to the LEI of Commerzbank International S.A.
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVH700~~	~~Reject File~~	~~Missing value for field Reporting Agent.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
~~Error~~	~~EVH701~~	~~Reject File~~	~~Invalid LEI format [REPORTING AGENT value] provided for field Reporting Agent.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH702	Reject File	Specified LEI [REPORTING AGENT value] does not match any of the Reporting Agents from the Reporting Population of [PANEL BANK CODE value] Panel Bank.	Synchronous
Error	EVH703	Reject File	Reporting Agent [REPORTING AGENT value] is different from the Reporting Agent indicated in the Message File Name.	Synchronous

3.3.2. Reference Period

Variable ID	Variable Name	Type	Description	Example
H80	REFERENCE PERIOD	Date-time [ISO 8601] YYYY-MM-DDThh:mm:ss +/-hh:mm	This is the start and end date and time of the period to which the data in the file refers. For Transaction data: it is the trade date for new transactions and the date of amendment, correction or cancellation; For Submission data: it is the TARGET date to which the submission (or quote) refers. In this case, the reported From and To dates must be identical. *The time zone information ('+/-hh:mm') must always be included for transaction data.*	From: 2016-07-01T18:00:00+01:00 To: 2016-07-02T18:00:00+01:00

Category	Code	Action	Description of validation	Timing
~~Error~~	~~EVH800~~	~~Reject File~~	~~Missing value for field Reference Period.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
~~Error~~	~~EVH801~~	~~Reject File~~	~~Invalid date format [Reference Period value] detected for field Reference Period.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~
Error	EVH802	Reject File	REFERENCE PERIOD <begin> [REFERENCE PERIOD <begin> value] is after REFERENCE PERIOD <end> [REFERENCE PERIOD <end> value]. --- *Note: This validation does not apply to Submission data reporting.*	Synchronous
Error	EVH803	Reject File	REFERENCE PERIOD <begin> [REFERENCE PERIOD <begin> value] and REFERENCE PERIOD <end> [REFERENCE PERIOD <end> value] are not identical where MARKET SEGMENT IDENTIFIER is Submission data. --- *Note: This validation does not apply to Transaction data reporting.*	Synchronous
Error	EVH804	Reject File	REFERENCE PERIOD <begin> [REFERENCE PERIOD <begin> value] is not today's date. Please submit data only for today's benchmark calculation. --- *Note: This validation does not apply to Transaction data reporting.*	Synchronous

3.4. Reporting Message for Submission Data

The following sections describe the data fields that are to be repeated for each reported EURIBOR quote:

3.4.1. Tenor

Variable ID	Variable Name	Type	Description	Example
Q10	TENOR	String. Max total length: 10 (see Code List)	Contains the information to identify the tenor for the submission. Valid Values For EURIBOR Code List: Description 1W: One-week tenor 1M: One-month tenor 3M: Three-month tenor 6M: Six-month tenor 12M: Twelve-month tenor	"1W" for the one-week tenor

Category	Code	Action	Description of validation	Timing
Error	EVQ100	Reject Submission	Missing value for field TENOR.	Asynchronous
Error	EVQ101	Reject Submission	Invalid value [TENOR value] provided for TENOR.	Asynchronous
Error	EVQ102	Reject Submission	Duplicate value [TENOR value] detected for field TENOR in one submitted file.	Asynchronous

3.4.2. Volume

Variable ID	Variable Name	Type	Description	Example
Q20	VOLUME	Numeric. Max total length: 15 Positive number.	This is the amount of money in **million euros** transacted at this tenor. Optional for EURIBOR.	1234 EUR for a volume of 1,234,000,000 euros. The reported currency must always be in euro ('EUR').

		Decimals: 0 Currency [ISO 4127]	**This field is mandatory for the EONIA benchmark. If provided for EURIBOR, this field will be ignored by EMMI – it will not be stored nor validated.**	

3.4.3. Rate

Variable ID	Variable Name	Type	Description	Example
Q30 	RATE	Numeric. Max total length: 11 Positive or negative number. Decimals: EURIBOR (2) Unit: Percentage points	This is the interest rate, expressed in accordance with the ACT/360 money market convention, at which the funding is estimated (EURIBOR quote). **This field is mandatory for EURIBOR.**	EURIBOR: -0.12 for a rate of minus 0.12%

Category	Code	Action	Description of validation	Timing
Error	EVQ300	Reject Submission	Missing value for field RATE.	Asynchronous
Error	EVQ301	Reject Submission	Invalid numeric format [RATE value] for field RATE.	Asynchronous

3.4.4. Submitter Identifier

Variable ID	Variable Name	Type	Description	Example
Q40	SUBMITTER IDENTIFIER	String Max total length: 35 [a-z][0-9]	A unique User ID used within the EBASS for the individual who submitted the record for processing. It is generated by the EBASS and composed of the user's first initials (i.e. 2 lowercase letters) followed by 4 random digits. **This field is mandatory for EURIBOR.** Test user IDs can be provided upon request.	cc4987

Category	Code	Action	Description of validation	Timing
Error	EVQ400	Reject Submission	Missing value for field SUBMITTER IDENTIFIER.	Asynchronous
Error	EVQ401	Reject Submission	User [SUBMITTER IDENTIFIER value] is not authorized as SUBMITTER for specified LEI [SENDER value] and Benchmark [MARKET SEGMENT IDENTIFIER value].	Asynchronous

3.4.5. Approver Identifier

Variable ID	Variable Name	Type	Description	Example
Q50	APPROVER IDENTIFIER	String Max total length: 35 [a-z][0-9]	A unique User ID used within the EBASS for the individual who approved the record for processing. It is generated by the EBASS and composed of the user's first initials (i.e. 2 lowercase letters) followed by 4 random digits. **This field is mandatory for EURIBOR.** Test user IDs can be provided upon request.	ql5927

Category	Code	Action	Description of validation	Timing
Error	EVQ500	Reject Submission	Missing value for field APPROVER IDENTIFIER.	Asynchronous
Error	EVQ501	Reject Submission	User [APPROVER IDENTIFIER value] is not authorized as APPROVER for specified LEI [SENDER value] and Benchmark [MARKET SEGMENT IDENTIFIER value].	Asynchronous
Error	EVQ502	Reject Submission	User [APPROVER IDENTIFIER value] approving the record is the same as User [SUBMITTER IDENTIFIER value] submitting the record – 4-eye principle violated.	Asynchronous

3.4.6. Reason Code

Variable ID	Variable Name	Type	Description	Example
Q60 	REASON CODE	String Max total length: [6]	This variable identifies the data source category underlying a Level 3 submission. For the current EURIBOR methodology, senders should use the default code "RC0000". Code List / Description RC0000 – Legacy Methodology RC0100 – Unused Transactions RC0200 – Executable Quotes RC0300 – Indicative Prices RC0400 – Internal Pricing RC0500 – Related Markets Data RC0600 – Other/Combination of above **This field is mandatory for EURIBOR.**	RC0200

Category	Code	Action	Description of validation	Timing
Error	EVQ600	Reject Submission	Missing value for field REASON CODE.	Asynchronous
Error	EVQ601	Reject Submission	REASON CODE [REASON CODE value] is not in the accepted list of values.	Asynchronous

3.4.7. Comment

Variable ID	Variable Name	Type	Description	Example
Q70 	COMMENT	String Max total length: 500	This variable provides the detailed rationale for a Level 3 submission. Instructions on how to construct the COMMENT are specified by EMMI separately.	

			This field is mandatory for EURIBOR only under the new Hybrid Methodology; it should not be provided for the current EURIBOR methodology. However, if provided this field will be stored and validated but not used.	
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVQ700	Reject Submission	Missing value for field COMMENT when the REASON CODE is not RC0000.	Asynchronous
Warning	EVQ701	No action	Content of field COMMENT is smaller than 10 characters.	Asynchronous

3.5. Reporting Message for Transaction Data

3.5.1. Which transactions to include?

- At a minimum, Panel Banks must report to EMMI all eligible transactions as described in the **Hybrid Methodology for the Determination of EURIBOR** (document version 1.1 or any subsequent revision).
- At most, Panel Banks may report to EMMI all MMSR-eligible transactions reported under the unsecured market segment.

The following are the fields used in the Reporting Message section for EURIBOR transaction data:

3.5.2. Data Set Action

This field is used only if there are no transactions to report.

Variable ID	Variable Name	Type	Description	Example
D10	DATA SET ACTION	String. Length: 4	This variable specifies the content of the message and triggers the appropriate processing in the receiving business application. 'NOTX' – The reporting agent has no activity to report in the market segment. This field is optional. If transactions are reported the report does not include this field.	NOTX
Category	**Code**	**Action**	**Description of validation**	**Timing**
~~Error~~	~~EVD100~~	~~Reject File~~	~~Invalid code [DATA SET ACTION value] provided for field DATA SET ACTION.~~ *[This validation is a part of the XSD validations]*	~~Synchronous~~

The following sections are to be repeated for each reported transaction:

3.5.3. Reported Transaction Status

Variable ID	Variable Name	Type	Description	Example

<table>
<tr><td>U10</td><td>REPORTED TRANSACTION STATUS</td><td>String. Length: 4

(see Code List)</td><td>This variable contains information about the status of the transaction, i.e. it includes details on whether the transaction is a new transaction, an amendment of a previously reported transaction, a cancellation of a previously reported transaction or a correction to a previously reported and rejected transaction.

<table><tr><th>Code List</th><th>Description</th></tr><tr><td>AMND</td><td>Amendment – transaction with this status will replace the previous version of the transaction.</td></tr><tr><td>CANC</td><td>Cancellation – transaction with this status (which had been previously reported with a status NEWT) will be removed from the processing queue.</td></tr><tr><td>CORR</td><td>Correction – transaction with this status will replace the previous version of the transaction.</td></tr><tr><td>NEWT</td><td>New transaction – this is how all transactions should be reported for the first time.</td></tr></table></td><td>NEWT</td></tr>
<tr><th>Category</th><th>Code</th><th>Action</th><th>Description of validation</th><th>Timing</th></tr>
<tr><td>Error</td><td>EVU100</td><td>Reject Transaction</td><td>Missing value for field REPORTED TRANSACTION STATUS.</td><td>Asynchronous</td></tr>
<tr><td>Error</td><td>EVU101</td><td>Reject Transaction</td><td>Invalid code [REPORTED TRANSACTION STATUS value] provided for field REPORTED TRANSACTION STATUS.</td><td>Asynchronous</td></tr>
<tr><td>Error</td><td>EVU102</td><td>Reject Transaction</td><td>REPORTED TRANSACTION STATUS = AMND where [PROPRIETARY TRANSACTION IDENTIFICATION value] is not present in database or is flagged as inactive due to a previous cancellation</td><td>Asynchronous</td></tr>
<tr><td>Error</td><td>EVU103</td><td>Reject Transaction</td><td>REPORTED TRANSACTION STATUS = CORR where [PROPRIETARY TRANSACTION IDENTIFICATION value] is not present in database or is flagged as inactive due to a previous cancellation</td><td>Asynchronous</td></tr>
<tr><td>Error</td><td>EVU104</td><td>Reject Transaction</td><td>REPORTED TRANSACTION STATUS = CANC where [PROPRIETARY TRANSACTION IDENTIFICATION value] is not present in database or is flagged as inactive due to a previous cancellation</td><td>Asynchronous</td></tr>
</table>

3.5.4. Novation Status

Variable ID	Variable Name	Type	Description	Example
U15	NOVATION STATUS	String. Length: 4 (see Code List)	This variable specifies whether the transaction is a novation, i.e. transactions in which the counterparty, inter alia, is changed. *The reporting is mandatory where applicable.* Code List: NONO – Transaction is not a novation; NOVA – Transaction is a novation	"NONO" is used for a transaction that is not a novation

Category	Code	Action	Description of validation	Timing
Warning	EVU150	No action	Missing value for field NOVATION STATUS.	Asynchronous
Error	EVU151	Reject Transaction	Invalid code [NOVATION STATUS value] provided for field NOVATION STATUS.	Asynchronous

3.5.5. Unique Transaction Identifier

Variable ID	Variable Name	Type	Description	Example
U20	UNIQUE TRANSACTION IDENTIFIER	String. Max Length: 105 Up to 105 alphanumerical characters. Four special characters are allowed ':', '.', '-', '_', but special characters not allowed at the beginning or the end. No space allowed.	The variable specifies the UTI, which is a unique code that allows the identification of a transaction in the respective market segment. **If provided, this field is ignored by EMMI – it will not be stored nor validated.**	

Category	Code	Action	Description of validation	Timing
Not validated	N/A	N/A	N/A	N/A

3.5.6. Proprietary Transaction Identifier

Variable ID	Variable Name	Type	Description	Example
U30	PROPRIETARY TRANSACTION IDENTIFIER	String. Max Length: 105	This is the unique internal transaction identifier used by the reporting agent for each transaction. The PTI with which each transaction will be transmitted and identified must be unique per market segment and reporting agent.	
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU300	Reject Transaction	Missing value for field PROPRIETARY TRANSACTION IDENTIFICATION.	Asynchronous
Error	EVU301	Reject Transaction	Duplicate value [PROPRIETARY TRANSACTION IDENTIFICATION value] detected in database for the same reporting agent and segment under the field PROPRIETARY TRANSACTION IDENTIFICATION where REPORTED TRANSACTION STATUS is NEWT.	Asynchronous
Error	EVU302	Reject Transaction	Duplicate value [PROPRIETARY TRANSACTION IDENTIFICATION value] detected for field PROPRIETARY TRANSACTION IDENTIFICATION in one submitted file.	Asynchronous
Error	EVU303	Reject Transaction	Invalid format [PROPRIETARY TRANSACTION IDENTIFICATION value] for PROPRIETARY TRANSACTION IDENTIFICATION	Asynchronous

3.5.7. Related Proprietary Transaction Identification

Variable ID	Variable Name	Type	Description	Example
U35	RELATED PROPRIETARY TRANSACTION IDENTIFICATION	String. Max Length: 105	This variable is the unique internal transaction identifier used by the reporting agent for the initial trade that was subsequently novated. *The reporting is mandatory where applicable.*	
Category	**Code**	**Action**	**Description of validation**	**Timing**

Warning	EVU350	No action	RELATED PROPRIETARY TRANSACTION IDENTIFICATION is provided when NOVATION STATUS is NONO.	Asynchronous
Warning	EVU351	No action	RELATED PROPRIETARY TRANSACTION IDENTIFICATION is not provided when NOVATION STATUS is NOVA.	Asynchronous
Warning	EVU352	No action	RELATED PROPRIETARY TRANSACTION IDENTIFICATION [RELATED PROPRIETARY TRANSACTION IDENTIFICATION value] is provided and not present in database for the same reporting agent and segment under the field PROPRIETARY TRANSACTION IDENTIFICATION.	Asynchronous

3.5.8. Counterparty Proprietary Transaction

Variable ID	Variable Name	Type	Description	Example
U40 	COUNTERPARTY PROPRIETARY TRANSACTION IDENTIFICATION	String. Max Length: 105	This variable specifies the PTI assigned by the counterparty of the reporting agent to the same transaction. **If provided, this field is ignored by EMMI – it will not be stored nor validated.**	
Category	**Code**	**Action**	**Description of validation**	**Timing**
Not validated	N/A	N/A	N/A	N/A

3.5.9. Counterparty Identification

3.5.9.1. Counterparty LEI

Variable ID	Variable Name	Type	Description	Example
U50 	COUNTERPARTY IDENTIFICATION	String. Length: 20	This variable provides the LEI of the counterparty of the reporting agent. *This variable is named 'LEI' in the message and located in the 'CounterpartyIdentification' block of the message.* **This field is not accepted by EMMI – variable U60 must be reported instead.**	
Category	**Code**	**Action**	**Description of validation**	**Timing**

Not validated	N/A	N/A	N/A	N/A

3.5.9.2. Counterparty Sector

Variable ID	Variable Name	Type	Description	Example
U60 	COUNTERPARTY SECTOR	String. Max Length: 4 Refers to ESA 2010 institutional sectors (see Code List)	This variable provides the institutional sector, e.g. non-financial corporation, central bank, etc. of the counterparty. **Unlike in MMSR, this field is mandatory for EMMI:** The COUNTERPARTY SECTOR must be provided **for all transactions** *where the COUNTERPARTY IDENTIFICATION is not provided.* *This variable is named 'Sector' in the message and located in the 'SectorAndLocation' block of the 'CounterpartyIdentification' block of the message.*	"S123" stands for "Money market funds"

Code List	Description
S11	Non-financial corporations
S121	Central bank
S122	Deposit-taking corporations except the central bank
S123	Money market funds
S124	Non-MMF investment funds
S125	Other financial intermediaries, except insurance corporations and pension funds
S126	Financial auxiliaries
S127	Captive financial institutions and money lenders
S128	Insurance corporations
S129	Pension funds
S13	General government

Category	Code	Action	Description of validation	Timing
Error	EVU600	Reject Transaction	Missing value for field COUNTERPARTY SECTOR ~~where COUNTERPARTY IDENTIFICATION is not provided.~~	Asynchronous

Not validated	EVU601	N/A	~~Missing value for field COUNTERPARTY SECTOR where COUNTERPARTY LOCATION [COUNTERPARTY LOCATION] is provided.~~	N/A
Error	EVU602	Reject Transaction	Invalid code [COUNTERPARTY SECTOR value] provided for field COUNTERPARTY SECTOR.	Asynchronous
Not validated	EVU604	N/A	~~COUNTERPARTY SECTOR provided as "S122".~~	N/A
Error	EVU605	Reject Transaction	COUNTERPARTY SECTOR is provided and different from "S122", where TRANSACTION TYPE provided as "LEND".	Asynchronous

3.5.9.3. Counterparty Location

Variable ID	Variable Name	Type	Description	Example
U70 	COUNTERPARTY LOCATION	String. Length: 2 [ISO3166-1 alpha-2]	This is the ISO country code of the country in which the counterparty is incorporated. ~~The COUNTERPARTY LOCATION must be provided for all transactions where the COUNTERPARTY IDENTIFICATION is not provided.~~ **If provided, this field is ignored by EMMI – it will not be stored nor validated.** *This variable is named 'Location' in the message and located in the 'SectorAndLocation' block of the 'CounterpartyIdentification' block of the message.*	DE refers to Germany
Category	**Code**	**Action**	**Description of validation**	**Timing**
Not validated	N/A	N/A	N/A	N/A

3.5.10. Trade Date

Variable ID	Variable Name	Type	Description	Example
U80	TRADE DATE	Date-time [ISO 8601] or	This variable specifies the date and time at which the parties enter into the reported transaction. It is to be reported with only the date when the time of the transaction is not available.	2017-12-22T09:00:00+00:00 refers to 22 December 2017 at 9:00 in GMT

		Date [ISO 8601]	The reported time is the execution time when available or alternatively the time at which the transaction entered the trading system of the reporting agent. The time must always reflect a real point in time and not be reported as a default value (e.g. midnight). The TRADE DATE must always equal to or be set before SETTLEMENT DATE. The only exception is in the case of novations, where TRADE DATE can be reported after SETTLEMENT DATE. For Date-time: YYYY-MM-DDThh:mm:ss+/-hh:mm or YYYY-MM-DDThh:mm:ss.sss+/-hh:mm The time zone information ('+/-hh:mm') must always be included. No local time format (i.e. without '+/-hh:mm') is allowed. For Date: YYYY-MM-DD The Date must always correspondent to the CET time zone.	Or 2017-12-22 stands for 22 December 2017 in CET
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU800	Reject Transaction	Missing value for field TRADE DATE.	Asynchronous
Error	EVU801	Reject Transaction	Invalid date format [TRADE DATE value] detected for field TRADE DATE.	Asynchronous
Error	EVU802	Reject Transaction	TRADE DATE [TRADE DATE value] is after SETTLEMENT DATE [SETTLEMENT DATE value].	Asynchronous
Error	EVU803	Reject Transaction	TRADE DATE [TRADE DATE value] is after MATURITY DATE [MATURITY DATE value].	Asynchronous
Error	EVU804	Reject Transaction	TRADE DATE [TRADE DATE value] is after CREATION DATE [CREATION DATE value]	Asynchronous

Error	EVU805	Reject Transaction	TRADE DATE [TRADE DATE value] is after end of REFERENCE PERIOD [REFERENCE PERIOD end value]	Asynchronous

3.5.11. Settlement Date

Variable ID	Variable Name	Type	Description	Example
U90	SETTLEMENT DATE	Date [ISO 8601] YYYY-MM-DD The Date must always correspondent to the CET time zone.	This is the date on which the amount of money is exchanged by counterparties or on which the purchase or sale of a debt instrument settles. With regard to call accounts and other unsecured borrowing/lending redeemable at notice, it is the date on which the deposit is rolled over, i.e. on which it would have been paid back if it had been called/not rolled over. In the case of a settlement failure in which settlement takes place on a different date than initially agreed, no transactional amendment needs to be reported.	2017-12-22 stands for 22 December 2017 in CET

Category	Code	Action	Description of validation	Timing
Error	EVU900	Reject Transaction	Missing value for field SETTLEMENT DATE.	Asynchronous
Error	EVU901	Reject Transaction	Invalid date format [SETTLEMENT DATE value] provided for field SETTLEMENT DATE.	Asynchronous
Error	EVU903	Reject Transaction	SETTLEMENT DATE [SETTLEMENT DATE value] is equal or after MATURITY DATE [MATURITY DATE value].	Asynchronous
Error	EVU904	Reject Transaction	SETTLEMENT DATE [SETTLEMENT DATE value] is after FIRST CALL/PUT DATE [FIRST CALL/PUT DATE value].	Asynchronous

3.5.12. Maturity Date

Variable ID	Variable Name	Type	Description	Example
U100	MATURITY DATE	Date [ISO 8601] YYYY-MM-DD The Date must always correspondent to the CET time zone.	The date on which the amount of money is due to be repaid by the borrower to the lender or on which a debt instrument matures and is due to be paid back. As regards callable and puttable instruments, the final maturity date must be provided. For call accounts and other unsecured borrowing/lending redeemable at notice, the maturity date must always be overnight. As an exception, in case the instrument cannot be redeemed/terminated/closed at overnight maturity, the first date on which it may be redeemed must be provided as maturity date. For saving accounts, the maturity should reflect the notice period of the account.	2017-12-22 stands for 22 December 2017 in CET

Category	Code	Action	Description of validation	Timing
Error	EVU1000	Reject Transaction	Missing value for field MATURITY DATE.	Asynchronous
Error	EVU1001	Reject Transaction	Invalid date format [MATURITY DATE value] provided for field MATURITY DATE.	Asynchronous
Error	EVU1004	Reject Transaction	The difference between MATURITY DATE [MATURITY DATE value] and SETTLEMENT DATE [SETTLEMENT DATE value] is more than 397 days.	Asynchronous
Error	EVU1005	Reject Transaction	MATURITY DATE [MATURITY DATE value] is before FIRST CALL/PUT DATE [FIRST CALL/PUT DATE value].	Asynchronous

3.5.13. Instrument Type

Variable ID	Variable Name	Type	Description	Example
U110	INSTRUMENT TYPE	String. Length: 4 (see Code List)	This variable identifies the instrument via which the borrowing/lending takes place. The instrument type will be selected from the reference table provided below: Code List: DPST – Deposit; CACM – Call account/call money	"DPST" stands for "deposit"

				CEOD	Certificate of deposit	
				COPR	Commercial paper	
				ABCP	Asset backed commercial paper	
				FRNT	Floating rate note	
				OTHR	Other short-term debt securities issued	

Category	Code	Action	Description of validation	Timing
Error	EVU1100	Reject Transaction	Missing value for field INSTRUMENT TYPE.	Asynchronous
Error	EVU1101	Reject Transaction	Invalid code [INSTRUMENT TYPE value] provided for field INSTRUMENT TYPE.	Asynchronous
Warning	EVU1102	No Action	INSTRUMENT TYPE is CACM where Call/Put Option is provided.	Asynchronous
Warning	EVU1103	No Action	INSTRUMENT TYPE is CEOD where RATE TYPE is VARI.	Asynchronous
Error	EVU1104	Reject Transaction	INSTRUMENT TYPE is FRNT where RATE TYPE is FIXE.	Asynchronous

3.5.14. Transaction Type

Variable ID	Variable Name	Type	Description	Example
U120	TRANSACTION TYPE	String. Length: 4 (see Code List)	This variable specifies whether the transaction is cash borrowing or cash lending. Code List: BORR = Borrowing cash; LEND = Lending cash	"BORR" stands for "borrowing"
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1200	Reject Transaction	Missing value for field TRANSACTION TYPE.	Asynchronous
Error	EVU1201	Reject Transaction	Invalid code [TRANSACTION TYPE value] provided for field TRANSACTION TYPE.	Asynchronous
Not validated	EVU1202	N/A	~~TRANSACTION TYPE = lend where [COUNTERPARTY SECTOR] is provided~~	N/A
Not validated	EVU1203	N/A	~~TRANSACTION TYPE = lend where [COUNTERPARTY LOCATION] is provided~~	N/A

3.5.15. Transaction Nominal Amount

Variable ID	Variable Name	Type	Description	Example
U130	TRANSACTION NOMINAL AMOUNT	Numeric. Max total length: 18 Positive number Decimals: up to 5 Currency [ISO 4127]	This is the amount of money in euro lent or borrowed on deposit. In the case of debt securities, it is the nominal amount of the security issued/purchased. The message must always specify that the currency is euro.	1000000 EUR, where 'EUR' stands for euro.
Category	**Code**	**Action**	**Description of validation**	**Timing**

Error	EVU1300	Reject Transaction	Missing value for field TRANSACTION NOMINAL AMOUNT.	Asynchronous
Error	EVU1301	Reject Transaction	Invalid numeric format [TRANSACTION NOMINAL AMOUNT value] provided for field TRANSACTION NOMINAL AMOUNT.	Asynchronous
Error	EVU1302	Reject Transaction	TRANSACTION NOMINAL AMOUNT [TRANSACTION NOMINAL AMOUNT value] is <= 0.	Asynchronous
Error	EVU1303	Reject Transaction	The currency specified for TRANSACTION NOMINAL AMOUNT [CURRENCY value] is not EUR.	Asynchronous

3.5.16. Transaction Deal Price

Variable ID	Variable Name	Type	Description	Example
U140	TRANSACTION DEAL PRICE	Numeric. Max total length: 11 Decimals: up to 10 Unit: Percentage points	This field contains the dirty price at which the security is issued or traded in percentage points, and which is to be reported as 100 for unsecured deposits. *This variable is named 'DealPrice' in the message.*	99.234 for transaction deal price of 99.234%
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1400	Reject Transaction	Missing value for field TRANSACTION DEAL PRICE.	Asynchronous
Error	EVU1401	Reject Transaction	Invalid numeric format [TRANSACTION DEAL PRICE value] provided for field TRANSACTION DEAL PRICE.	Asynchronous
Warning	EVU1402	No action	TRANSACTION DEAL PRICE [TRANSACTION DEAL PRICE value] =! 100 where INSTRUMENT TYPE is DPST or CACM.	Asynchronous
Error	EVU1403	Reject Transaction	TRANSACTION DEAL PRICE [TRANSACTION DEAL PRICE value] is less than 0.	Asynchronous

3.5.17. Rate Type

Variable ID	Variable Name	Type	Description	Example
U150	RATE TYPE	String. Length: 4 (see Code List)	This indicates the type of rate applicable for the transaction. Possible values: • fixed rate for deposits and debt instruments with fixed coupons; • variable rate for debt instruments and unsecured deposits for which the pay out at maturity or period depends on the observed value of some underlying reference rate. Code List: FIXE = Fixed rate; VARI = Variable rate	"FIXE" stands for a "fixed rate"
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1500	Reject Transaction	Missing code for field RATE TYPE.	Asynchronous
Error	EVU1501	Reject Transaction	Invalid code [TYPE OF RATE value] provided for field RATE TYPE.	Asynchronous

3.5.18. Deal Rate

Variable ID	Variable Name	Type	Description	Example
U160	DEAL RATE	Numeric. Max total length: 11 Positive or negative number Decimals: up to 10 Unit: Percentage points	This is the interest rate, expressed in accordance with the ACT/360 money market convention, at which the deposit was concluded and at which the cash amount lent is remunerated. In the case of debt instruments, this is the effective interest rate, expressed in accordance with the ACT/360 money market convention, at which the instrument was issued or purchased. This value can be positive or negative irrespective of whether the cash is borrowed or lent. It represents the contractually agreed remuneration rate on the transaction nominal amount regardless of the transaction sign (i.e. whether the TRANSACTION TYPE is borrowed or lent).	10.234 for a deal rate of 10.234% And -10.234 for a deal rate of minus 10.234%

			This field will only be reported in case RATE TYPE is fixed rate (FIXE) or when reporting a fixed-rate equivalent in variable rate (VARI) transactions.	
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1600	Reject Transaction	Missing value for field DEAL RATE when RATE TYPE is FIXE or VARI with REFERENCE RATE INDEX equal to the unsecured euro overnight reference rate.	Asynchronous
Error	EVU1601	Reject Transaction	Invalid numeric format [DEAL RATE value] provided for field DEAL RATE.	Asynchronous
Error	EVU1602	Reject Transaction	DEAL RATE [DEAL RATE] provided when RATE TYPE is VARI and REFERENCE RATE INDEX is not the unsecured euro overnight reference rate.	Asynchronous

3.5.19. Reference Rate Index

Variable ID	Variable Name	Type	Description	Example
U170	REFERENCE RATE INDEX	String. Length: 12 ISIN of the underlying [ISO6166] First two characters must be alphabetic (A-Z)	This variable provides the ISIN code of the underlying reference rate on the basis of which the periodic interest payments are calculated. A complete list of applicable ISIN codes for the different REFERENCE RATE INDICES is available in Annex IV of the MMSR Reporting instructions2. *This field will only be reported for floating rate instruments.* *This variable is located in the 'FloatingRateNote' block of the message.*	'EU0009652783' stands for the 3 month EURIBOR.
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1700	Reject Transaction	Missing value for field REFERENCE RATE INDEX where RATE TYPE is VARI.	Asynchronous
Error	EVU1701	Reject Transaction	REFERENCE RATE INDEX [REFERENCE RATE INDEX] provided when RATE TYPE is FIXE.	Asynchronous
Error	EVU1702	Reject Transaction	Invalid identifier format [REFERENCE RATE INDEX value] provided for field REFERENCE RATE INDEX.	Asynchronous

[2] https://www.ecb.europa.eu/stats/financial_markets_and_interest_rates/money_market/html/index.en.html

3.5.20. Basis Point Spread

Variable ID	Variable Name	Type	Description	Example
U180	BASIS POINT SPREAD	Numeric. Max total length: 18 Positive or negative number Decimals: 0 Unit: Basis points	The number of basis points added to (if positive) or deducted from (if negative) the reference rate index to calculate the actual interest rate applicable for a given period at issuance of the floating rate instrument. *This field will only be reported for floating rate instruments.* *This variable is located in the 'FloatingRateNote' block of the message.*	1023 for a spread of 10.234%. As the unit is in basis points, here the 10.234% is first rounded to 10.23% and then converted to 1023.

Category	Code	Action	Description of validation	Timing
Error	EVU1800	Reject Transaction	Missing value for field BASIS POINT SPREAD where RATE TYPE is VARI.	Asynchronous
Error	EVU1801	Reject Transaction	BASIS POINT SPREAD [BASIS POINT SPREAD value] provided where RATE TYPE is FIXE.	Asynchronous
Error	EVU1802	Reject Transaction	Invalid numeric format [BASIS POINT SPREAD value] provided for field BASIS POINT SPREAD.	Asynchronous

3.5.21. Call or Put

Variable ID	Variable Name	Type	Description	Example
U190	CALL OR PUT	String. Length: 4 (see Code List)	This variable identifies whether the instrument has a call option or a put option. If the instrument contains both options, i.e. a call and a put, both the call option and the put option have to be reported. Code List: CALL = Call; PUTO = Put *To be reported only for callable/puttable instruments.*	"CALL" stands for a call

			If the instrument is identified as callable or puttable, at least one of the fields FIRST CALL/PUT DATE and CALL/PUT NOTICE PERIOD must be reported. *This variable must not be reported for call account/call money and saving account transactions.* *This variable is located in the 'CallPutOption' block and named 'Type' in the message.*	
Category	**Code**	**Action**	**Description of validation**	**Timing**
Error	EVU1900	Reject Transaction	Missing value for field CALL OR PUT if FIRST CALL/PUT DATE [FIRST CALL/PUT DATE value] or CALL/PUT NOTICE PERIOD [CALL/PUT NOTICE PERIOD value] is provided.	Asynchronous
Error	EVU1901	Reject Transaction	Invalid code [CALL OR PUT value] provided for field CALL OR PUT if FIRST CALL/PUT DATE [FIRST CALL/PUT DATE value] or CALL/PUT NOTICE PERIOD [CALL/PUT NOTICE PERIOD value] is provided.	Asynchronous
Warning	EVU1902	No action	CALL or PUTO are provided twice for [CALL OR PUT] for the same transaction.	Asynchronous

For transactions with embedded call or put options, only one or the other of the following two variables must be reported – not both:

- Earliest Exercise Date (U200) or
- Notice Period (U210)

3.5.22. Earliest Exercise Date

Variable ID	Variable Name	Type	Description	Example
U200	FIRST CALL/PUT DATE	Date [ISO 8601] YYYY-MM-DD The Date must always correspondent to the CET time zone.	This is the first date on which the call option or the put option can be exercised. *This reporting is mandatory where applicable, i.e. for instruments with a call/put option that can be exercised on one or more predefined dates.* *This variable must not be reported for call account/call money and saving account transactions.* *This variable is located in the 'CallPutOption' block and named 'EarliestExerciseDate' in the message.*	2017-12-22 stands for 22 December 2017 in CET

Category	Code	Action	Description of validation	Timing
Error	EVU2000	Reject Transaction	Missing value for field FIRST CALL/PUT DATE if CALL/PUT [CALL/PUT value] is provided and CALL/PUT NOTICE PERIOD blank.	Asynchronous
Error	EVU2001	Reject Transaction	Invalid date format [FIRST CALL/PUT DATE value] provided for field FIRST CALL/PUT DATE.	Asynchronous

3.5.23. Notice Period

Variable ID	Variable Name	Type	Description	Example
U210	CALL/PUT NOTICE PERIOD	Numeric. Max total length: 18 Decimals: 0	This is the number of calendar days that the holder of the instrument/issuer of the instrument will give to the issuer/holder of the instrument before exercising the put/call option. *This reporting is mandatory where applicable, i.e. for all instruments/transactions with a call/put option notice period and for deposits redeemable at a pre-agreed notice period i.e. for all instruments where the option holder must provide a minimum number of days to the counterparties before the option can be exercised.* *This variable must not be reported for call account/call money and saving account transactions.* *This variable is located in the 'CallPutOption' block and named 'NoticePeriod' in the message.*	"7" stands for a notice period of one week

Category	Code	Action	Description of validation	Timing
Error	EVU2100	Reject Transaction	Missing value for field CALL/PUT NOTICE PERIOD if CALL/PUT [CALL/PUT value] is provided and FIRST CALL/PUT DATE blank.	Asynchronous
Error	EVU2101	Reject Transaction	Invalid numeric format [CALL/PUT NOTICE PERIOD value] provided for field CALL/PUT NOTICE PERIOD.	Asynchronous
Error	EVU2102	Reject Transaction	CALL/PUT NOTICE PERIOD [CALL/PUT NOTICE PERIOD value] is greater than the difference between SETTLEMENT DATE [SETTLEMENT DATE value] and MATURITY DATE [MATURITY DATE value].	Asynchronous

4. General Formats

4.1. Representation of numbers

In general, a dot (".") has to be used as a decimal separator and the reported values are not allowed to contain a comma as a thousand separator.

- For example, to represent a value of 1 million euro and 5 cents, the correct format would be "1000000.05". Representations like "1,000,000.05" or "1000000,05" will be considered invalid.

Integers must always be provided with exact precision and must not be rounded.

- For example, if the number of days is 358, it should not be rounded to 360. Rounding, if any, should be done by the receiver.

Decimals must be reported in line with the field definitions; furthermore, the maximal number of decimals available must be provided. Rounding, if any, should be done by the receiver.

4.2. LEI for EMMI

The Legal Entity Identifier (LEI) is a 20-digit, alphanumeric code that connects to key reference information that enables clear and unique identification of companies participating in global financial markets. The LEI is based on the ISO standard 17442 developed by the International Organization for Standardization. As defined in ISO standard 17442, any legal entity that enters into a financial transaction is eligible for an LEI.
To date, EMMI does not possess an LEI. Hence, for the purpose of this document only, we have created an artificial number which can be used as LEI for EMMI: **BE04650754O8EMMI0001.**

5. Data Validations

All data files reported by Panel Banks will undergo validation checks. The checks will be applied to the transmitted data to validate its format, quality and consistency. Panel Banks are advised to implement similar validation checks in their system to enhance the quality of their data and hence avoid possible processing delays.

There are two layers of data validations performed by the EBASS and visible to Panel Banks:

- **Synchronous validations** are performed upon receipt of the delivery message; they verify that the reported data is structured as per the expected technical standard and that key routing information in the Header section is accurate. Results of such checks are returned to Panel Banks in a synchronous response message. They will also be made available via the web-based GUI;
- **Asynchronous validations** are run within 10 minutes of the acceptance of the delivery message; they verify the quality and consistency of the reported transaction and submission data. Results of such checks are returned to Panel Bans only on demand, using the Feedback Service; they will also be made available via the web-based GUI.

The individual validation rules applicable to headers and record-based data are presented in a tabular format in sub-sections **3.2** to **0**. Validations related to the same variable are displayed in a distinct table. The tabular rules consist of five individual attributes:

1. The validation **Category**, which reflects the importance of the check and defines how the record will be treated;
2. The validation **Code**, which serves as a unique identification of a validation rule and is returned by the EBASS if the validation fails;
3. The data **Action**, which describes the action taken by the EBASS after application of a check, e.g. "reject transaction";
4. The validation **Description**, which is a textual description of the validation rule;
5. The validation **Timing**, corresponding to the point-in-time when the check is applied (i.e. synchronously or asynchronously);

The validation code is formatted to facilitate easy identification of the source and reason the data field was flagged; it is composed of three elements.

i. All validation codes share the same first 2 letters, 'EV' which stands for EMMI Validation;

ii. Following 'EV', the validation code consists of three (or four) characters identify the variable for which the check is applied. i.e. H10, H20, etc. for header fields; U10, U20, etc. for unsecured transaction data fields; i.e. Q10, Q20, etc. for submission data fields;

iii. Finally, a sequential number uniquely identifies the validation rule for a specific variable, i.e. 0, 1, 2, etc.

Examples of a Validation Code

EVH20**1** refers to the second (**1**) EMMI validation (**EV**) rule of the second variable (10) on the unsecured transaction (U).

EVU10**0** refers to the first (**0**) EMMI validation (**EV**) rule of the first variable (10) on the unsecured transaction (U).

The possible values of error category are:

- 'Error' – A blocking error, indicating that a record (or the entire file) will be rejected;
- 'Warning' – A non-blocking error, indicating the record will be accepted but the Panel Bank should investigate whether there is a mistake.

Important notes:

Although many of the validations defined herein are derived from the MMSR standards, some have been amended, removed, or added. In order to avoid confusion with the MMSR standards, EMMI uses the EV prefix as part of the EBASS solution to explicitly distinguish its data quality checks from those employed by the ECB.

Panel Banks should be aware that, in addition to the validation checks defined in this document, further validations may be carried out by EMMI to check data consistency and reasonableness. For these purposes, Panel Banks may be contacted to further clarify possible data quality issues or inconsistencies. These additional checks are not included in any status message. These will be dealt with by EMMI and Panel Banks on a case-by-case basis.

6. Web services

The benchmark data defined in section 3 above must be reported in a unified Extensible Markup Language (XML) format. Data will be transmitted to a single reception point via a secured internet-based application-to-application (A2A) channel. Data files will undergo the validation checks, described in the preceding section, when they are received by the EBASS, and automated status messages will be sent back to the sender. Data rejected by the automated validation checks must be corrected and resubmitted by the sender.

To transmit data and request status messages, senders must make web service calls using a SOAP open standard. Technical details of the submission and feedback flows are provided in this section. When an error occurs, a SOAP exception is returned to the sender. There can be no recovery without the intervention of the issuer. Thus, the sender is responsible for resending data in the event of a failure.

In parallel, Panel Bank users will be able to monitor the submitted data files and the status messages containing the results of the validation checks, as well as reports via a web-based interface or GUI.