EUROPEAN CENTRAL BANK

EUROSYSTEM

Money Markets Statistical Reporting (MMSR)

IT Appendix for Reporting Agents

Version	Status	Date
1.0	First version	08.09.2015
1.01	Updates	23.09.2015
1.02	Updates	20.11.2015
1.03	Updates	11.01.2016
3.0	Updates	15.12.2017
4.0	Updates (no change on requirements)	06.10.2023

Contents

1. Glossary

Term	Definition
A2A	Application-to-application submission process.
CAF	In the European context, CAF relates to the multi-acceptance of certificates which are compatible with the MMSR system.
ESCB	The European System of Central Banks (ESCB) is composed of the European Central Bank (ECB) and the national central banks (NCBs) of all 27 European Union (EU) Member States.
FX Swaps	Foreign exchange swaps, one of the four market segments reported under MMSR.
IAM	IAM is a shared ESCB service used to authenticate senders and manage their access rights for the Transactional Module.
LEI	Legal entity identifier.
MMSR system	The Money Market Statistical Reporting (MMSR) system comprises a Transactional Module and an Analytical Module. Only the Transactional Module is used by the senders and described in this document. The MMSR Transactional Module receives ISO 20022 XML files and returns automatic notifications to the sender.
OIS	Overnight index swaps, one of the money market segments reported under MMSR..
PKI	Public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling the sender to both securely exchange data with the MMSR system and prove its identity.
RA	Reporting agents (RAs) are commercial banks that report to the MMSR system. A commercial bank may be the sender of the submission or may delegate that to a third party.
Receiver	MMSR Transactional Module
Sender	The sending application.
SOAP	Simple Object Access Protocol, a message protocol that enables the distributed elements of an application to communicate.
Submission	File submission relates to the reception in the Transactional Module of a dataset sent by a sender.
TLS protocol	Transport layer security protocol.
U2A	User-to-application submission process.
UI	User interface.
UTF-8	Unicode (Universal Coded Character Set) Transformation Format – 8-bit, a variable-lengths character encoding standard used for electronic communication.

2. Introduction

The Money Markets Statistical Reporting solution is an IT application for the collection, storage, processing, compilation, and dissemination of money market data collected from credit institutions located in the euro area under Regulation (EU) No 1333/2014 of the European Central Bank of 26 November 2014 concerning statistics on money markets (ECB/2014/48). The main purpose of collecting such statistics is to provide the European Central Bank (ECB) with comprehensive, detailed and harmonised statistical information on the money markets in the euro area. The transaction data collected in respect of those markets provide information on the transmission of monetary policy decisions. The collection of statistical data is also necessary to enable the ECB to provide analytical and statistical support to the European Banking Supervision in accordance with Council Regulation (EU) No 1024/2013 of 15 October 2013 conferring specific tasks on the European Central Bank concerning policies relating to the prudential supervision of credit institutions.

Reporting agents are required to report to the ECB or the relevant national central bank (NCB) data on secured transactions, unsecured transactions, foreign exchange swaps and overnight index swaps. The actual reporting population consists of MFIs resident in the euro area that have been identified by the ECB Governing Council.

3. Scope of document

This document describes the submission and feedback processes to be followed by reporting agents (or the sender, in case of a delegated submission) reporting directly to the MMSR Transactional Module, i.e. reporting agents reporting to the ECB. Reporting agents reporting to the Deutsche Bundesbank, the Banco de España, or the Banca d'Italia should instead refer to the related documentation available from these institutions.
The document describes the prerequisites for the use of web services in Section 4 Prerequisites for the use of web services.

Additionally, it defines in Section 6 the technical requirements of the MMSR file to be submitted by the sender (RA) and it covers the two steps of a submission to the MMSR System:

1) Delivery Service (synchronous):
   The MMSR system provides information on the submission technical status and the DeliveryID to be used in the second step.
   The details are contained in Section 5.3 Details on the synchronous step of an MMSR submission and in Section 7 "ReceiveDeliveryService"_web_service.

2) Feedback Service (asynchronous)
   The MMSR system provides information on the submission business status.
   The details are in Section 5.4 Details on the asynchronous step of an MMSR submission and in Section 8 "GetFeedbackService"_web_service.

Finally, it briefly introduces the MMSR UI in Section 9 MMSR logging tool and the Availability of the MMSR system in Section 10 Availability of the MMSR system.

4. Prerequisites for the use of web services

MMSR system is protected by certificate authentication. Hence, the reporting agent should obtain a valid ESCB software certificate (not expired or revoked) delivered by ESCB-PKI.
For more details on how to obtain the ESCB software certificate, please contact Banque de France by email at: MMSR_MOA@banque-france.fr
Exchanges with MMSR system should be secured by the Transport Layer Security (TLS) 1.2 protocol.

5. Description of MMSR processes

The ECB collects daily transactional data on secured, unsecured, OIS and FX Swaps transactions in four separate files. The data are to be reported in a unified Extensible Markup Language (XML) format (ISO 20022-compliant) daily.

Data are submitted to a single reception point, either at the ECB or at the relevant NCB, via a secure transmission channel.

Data files undergo validation checks when they are received by the MMSR system, and automated status messages are made available to the sender via an A2A channel.

Reporting agents can also monitor the data files and the status messages containing the results of the validation checks, as well as reports via a UI through a web-based logging tool.

Data rejected by the automated validation checks must be corrected and resubmitted.

MMSR uses a secured internet-based A2A channel. Senders can make web service calls using an open SOAP standard.

Technical details of the submission and feedback flows are provided below.

5.1. Reporting data to MMSR

The submission of a report to MMSR from a reporting agent to the ECB takes place in two steps:

- The RA submits the data to be reported embedded in a delivery message to MMSR using the *ReceiveDeliveryService* web service (see in Section 7 “ReceiveDeliveryService” web service for technical details. The web service produces a synchronous response that includes a unique identifier (*DeliveryId*) and a technical status message containing a list of errors, if any, resulting from the technical validation of the submission (checking for corrupted files, checking authorisation, ensuring that file names comply with naming conventions, etc).

- Once the file has been processed by the system (around 5 minutes later), the RA can asynchronously call the *GetFeedbackService* web service (see in Section 8 “GetFeedbackService” web service for technical details) by quoting the *DeliveryId* previously obtained to obtain the results of the business validation of the information submitted. This will be contained in a business status message embedded in the webservice response. The business validation checks include checks for the ISO 20022 XML headers and for the individual transactions reported.

5.2. Business rules for sending RA delivery messages

- If the RA has no data to submit for a given market segment on a given day, it must still send to MMSR an RA delivery message for that segment with the DataSetAction XML tag populated with “NOTX”.

- The <BusinessService> XML tag in the ISO 20022 XML business application header can be populated with one of the following two values:

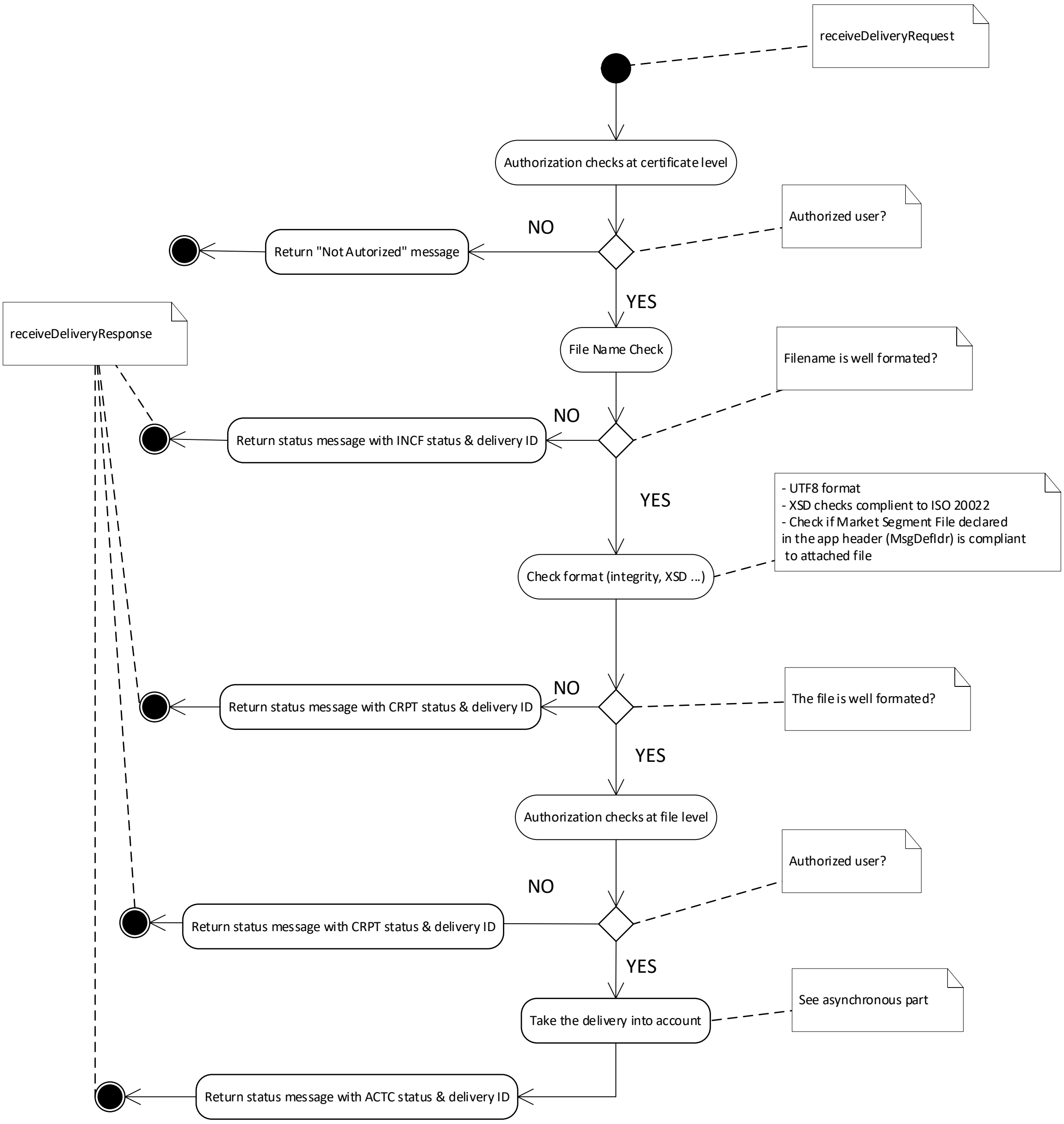
  (a) "ECB_MMSR_PROD" if the RA wishes to send a delivery (standard case);

  (b) "ECB_MMSR_TEST" if the RA wishes to test the technical channel. In that case, the MMSR will behave as follows:
    - The delivery process via the *ReceiveDeliveryService* will return a technical status message (except in case of fatal SOAP errors as explained below in Section 5.3 Details on the synchronous step of an MMSR submission.
    - However, no data will be stored in MMSR, no *DeliveryId* will be produced and no business status message is created.

Refer to Section 7.4 MMSR technical status message and to Annex I: Technical documentation for details on the XML structure of the MMSR delivery messages.

5.3. Details on the synchronous step of an MMSR submission

The synchronous step of an MMSR submission takes place via a call to the *ReceiveDeliveryService* web service. The system behaviour under different scenarios is detailed below using a flow chart:


receiveDeliveryRequest
Authorization checks at certificate level
Authorized user?
NO
Return "Not Autorized" message
YES
receiveDeliveryResponse
File Name Check
Filename is well formated?
NO
Return status message with INCF status & delivery ID
YES
- UTF8 format
- XSD checks complient to ISO 20022
- Check if Market Segment File declared in the app header (MsgDefIdr) is compliant to attached file
Check format (integrity, XSD ...)
NO
Return status message with CRPT status & delivery ID
The file is well formated?
YES
Authorization checks at file level
Authorized user?
NO
Return status message with CRPT status & delivery ID
YES
Take the delivery into account
See asynchronous part
Return status message with ACTC status & delivery ID

Details on the process are provided below:

1. Authorisation checks at certificate level:

   (a) If the user is not authorised to interact with MMSR, a generic “not authorised” message is returned (see Section 10.4 Technical errors).

   (b) If the user is authorised, the system proceeds to the next step.

2. Checks on the file name:

   (a) If the file name does not comply with the predefined file name conventions (see point 5 of Section 6 Technical requirements of the MMSR file for such conventions), a technical status message is returned with a technical status of “INCF”. In this case, the RA field of the technical status message is populated with the “unknown LEI” XXXXXXXXXXXXXXXXXX00 and the reporting period field is populated with the “unknown date” 9999-12-31. The reporting agent must therefore resubmit all the transactions within the rejected report, correcting the errors in the file name.

(b) If the file name complies with the file name conventions, the system proceeds to the next step.

3. Checks on the file format:

(a) If the file format is not compliant with ISO 20022 or is not a UTF-8 format, a technical status message is returned with a technical status of “CRPT”. In this case, the RA field of the technical status message is populated with the “unknown LEI” XXXXXXXXXXXXXXXXXX00 and the reporting period field is populated with the “unknown date” 9999-12-31. The reporting agent must therefore resubmit all the transactions within the rejected report, correcting the errors in the file format.

(b) If the file is formatted correctly, the system proceeds to the next step.

4. Checks on file information:

(a) If the *MessageDefinitionIdentifier* (MsgDefIdr) provided is different from the content of the report, or if the receiver’s LEI is not the ECB’s, a technical status message is returned with a technical status of “CRPT”. The reporting agent must therefore resubmit all the transactions within the rejected report, correcting the errors in the *MessageDefinitionIdentifier* or in the receiver’s LEI.

(b) If the sender is not authorised to deliver for a particular segment, a technical status message is returned with a technical status of “CRPT”. The reporting agent must therefore resubmit all the transactions within the rejected report (unless the submission was made in error) using the correct authorisation.

(c) Otherwise, the system proceeds to the next step.

5. Return the technical status message with a technical status of “ACTC”.

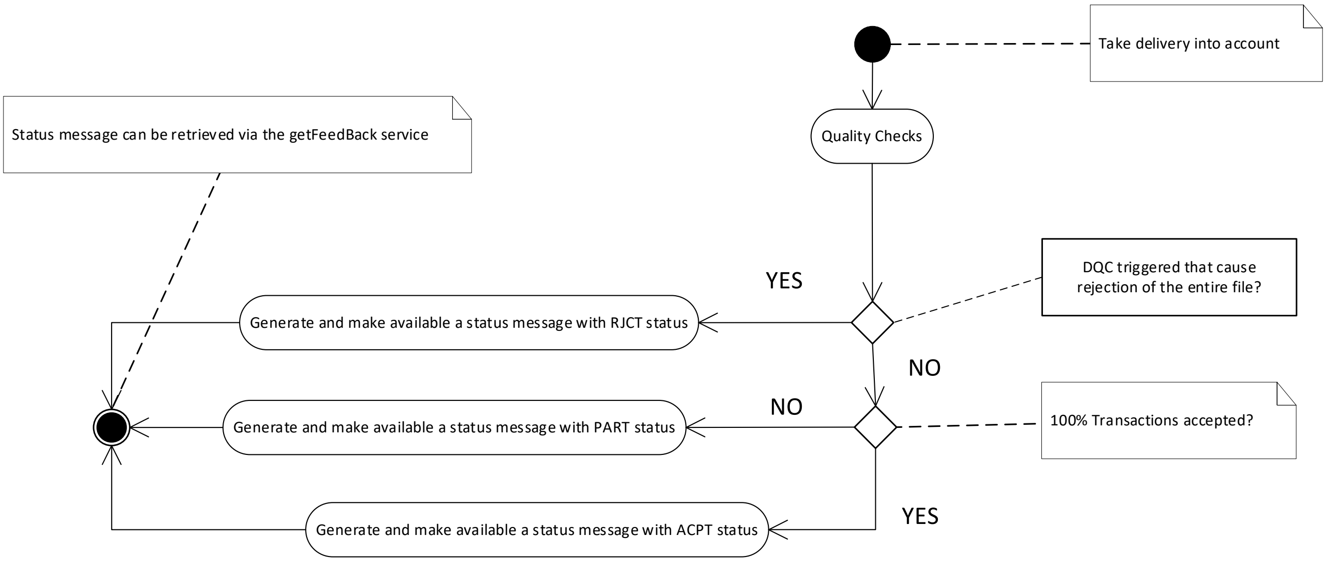
All files for which a technical status of “ACTC” is returned are taken into account by the system. This means that a series of business checks on the header and on the transactions reported in the file will be executed and a business status message will be prepared with their results. The message will be available asynchronously via the A2A *GetFeedbackService*.

Important: For security reasons, a generic “not authorised” message is returned by the service in the event of an unauthorised user. This message is not compliant with the ISO 20022 *StatusMessageFile* format.

Further technical details are provided in Section 7 “ReceiveDeliveryService” web service.

5.4. Details on the asynchronous step of an MMSR submission

The asynchronous step of an MMSR submission starts when a report has been accepted by the system with technical status “ACTC”. The system executes a series of business quality checks with results made available (after a delay) via a call to the *GetFeedbackService* web service. The system behaviour during this process under different scenarios is detailed below using a flow chart:



Take delivery into account
Status message can be retrieved via the getFeedBack service
Quality Checks
YES
DQC triggered that cause rejection of the entire file?
Generate and make available a status message with RJCT status
NO
NO
Generate and make available a status message with PART status
100% Transactions accepted?
YES
Generate and make available a status message with ACPT status

Details of the process are provided below:

1. A correctly formatted report that has cleared all the technical checks described Section 5.3 Details on the synchronous step of an MMSR submission is received by the system.

2. Quality checks are computed and a business status message for the report is generated and made available via the A2A *GetFeedbackService*. This message contains a business result status for the submitted ISO20022 report, which can be one of the following:

   (a) **“RJCT”** (report rejected). The submitted report is rejected due to errors in the report’s header. In this case, the transactions included in the report are not processed and their individual identifiers are not registered in the system’s database. The reporting agent must therefore resubmit all the transactions within the rejected report, correcting the errors in the header. The transactions should be re-sent using the same status as in the original submission.[1]

[1] In particular, transactions that were sent with status ‘NEWT’ in the rejected submission must be resent with that same status, their identifiers were not recorded by the system due to the complete rejection of the file. For an overview of the possible errors related to the life-cycle of transactions, refer to *Money Market Statistical Reporting (MMSR)* – Data Quality Checks document, checks with suffix 102,103,104,301 and 302 among those with prefix DQS,DQU,DQF and DQO.

(b) **“PART”** (report partially accepted). The submitted report is accepted but one or more of the transactions contain errors. In this case, the transactions included in the report are processed, and any new transaction identifiers are registered with the system. The reporting agent must resubmit only the rejected transactions using the transaction status “CORR”.[2]

(c) **“ACPT”** (report accepted). The submitted report is accepted and there are no errors in any of the transactions submitted.

In cases (b) and (c) above, the message also includes, where applicable, a block with the result of the quality checks for each transaction reported in the submission.

The technical details are provided in Section 8 *“GetFeedbackService” web service*. For details of the quality checks applied both to the report’s header and to the individual transactions, see the Money Market Statistical Reporting (MMSR) – Data Quality Checks document.

[2] This is because the identifiers of the rejected transactions in a report with “PART” status are registered with the system, so they can only be amended, corrected, or cancelled on re-submission. For an overview of the possible errors related to the life-cycle of transactions, refer to *Money Market Statistical Reporting (MMSR)* – Data Quality Checks document , checks with suffix 102,103,104,301 and 302 among those with prefix DQS,DQU,DQF and DQO.

6. Technical requirements of the MMSR file

Reporting Agents must submit MMSR ISO 20022 XML files to the MMSR system, as detailed in the Reporting Instructions.
There are certain technical requirements that should be considered while generating the XML files:

1) Structure

Each file submitted to the MMSR system is a business message relating to one of the four different market segments.

A business message for a particular market segment consists of two components:

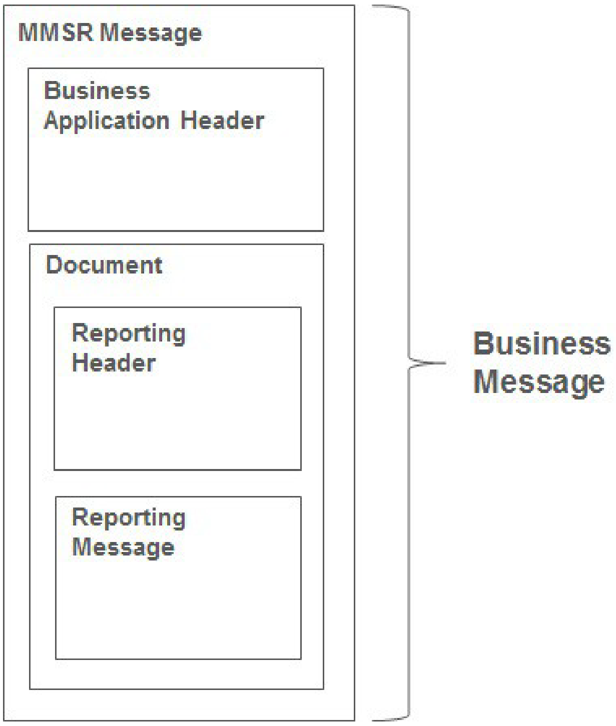
(a) a Business Application Header (BAH), which is used to identify the message and includes routing information;

(b) a document consisting of two parts: the Reporting Header and the Reporting Message for the specific market segment.

(i) The Reporting Header is used to identify the relevant reporting agent, the reference period and the overall content of the message.

(ii) The Reporting Message contains detailed information on transactions in the relevant market segment.

The diagram below depicts the conceptual structure of the MMSR message.



2) Technical wrapper

For the submission of the ISO 20022-compliant MMSR messages, a “technical wrapper” is used to convey the business message. This is a kind of technical envelope that allows the transmission of the Business Application Header and the Reporting Message.
Below is an example of a technical wrapper:

```
<?xml version="1.0" encoding="UTF-8"?>
<MMSRMessage
 xmlns:h="urn:iso:std:iso:20022:tech:xsd:head.001.001.01"
 xmlns:s="urn:iso:std:iso:20022:tech:xsd: auth.012.001.02"
 xmlns:u="urn:iso:std:iso:20022:tech:xsd: auth.013.001.02"
 xmlns:fx=" urn:iso:std:iso:20022:tech:xsd: auth.014.001.02"
 xmlns:ois="urn:iso:std:iso:20022:tech:xsd:
 auth.015.001.02">
   <h:AppHdr> ….
   </h:AppHdr>
   <s:Document> …
   </s:Document>
</MMSRMessage>
```

The XSD file can be found in Annexes (Annex I: Technical documentation)

3) Rules on sequencing of XML tags

The XML tags defined in the XSD file follow precise sequencing.

For example, in an “unsecured market” XML file, the <DealPric> tag must appear before the <RateTp> tag. If it appears after the <RateTp> tag, the file will be rejected by the MMSR system.

4) File size limitation

The limit for file size is fixed to 25Mb.

5) MMSR delivery file name conventions

In accordance with ISO 20022, MMSR messages consist of two components: a Business Application Header and a Reporting Message. Each MMSR message relates to a single reporting agent and a single market segment.

The names of MMSR messages must follow the pattern below:

<MARKET SEGMENT IDENTIFIER>.<LEI>.<DATE>.<INCREMENTAL TRANSMISSION NUMBER>

Variables	Description
<MARKET SEGMENT IDENTIFIER>	Length: 15 characters The market segment for which data are being submitted The four market segments are as follows: "auth.012.001.02" for secured markets; "auth.013.001.02" for unsecured markets; "auth.014.001.02" for foreign exchange swaps; "auth.015.001.02" for overnight index swaps.
<LEI>	Length: 20 characters Reporting agent's LEI
<DATE>	Length: 8 characters Reporting date Date of data as specified in ISO 8601 in the following format: YYYYMMDD
<INCREMENTAL TRANSMISSION NUMBER>	Length: 4 characters Incremental numerical variable The first file transmitted per segment per day will have the value "0001".

For example, the second secured market message transmitted by UniCredit Bank Austria AG on 16 January 2017 will have the following name: *auth.012.001.02.D1HEB8VEU6D9M8ZUXG17.20170116.0002*

7. "ReceiveDeliveryService" web service

The system provides a web service called the "ReceiveDeliveryService" to submit reports to the system.

This section describes the web service and provides a number of examples.

7.1. Description of the "ReceiveDeliveryService" web service

The ReceiveDeliveryService is a synchronous web service that allows the sender to submit ISO 20022 XML files to the MMSR system. It uses a secured A2A channel.

This web service is composed of a request (ReceiveReportingDeliveryRequest) and a response (ReceiveReportingDeliveryResponse).

The steps performed by the ReceiveReportingDeliveryRequest function are as follows:

- checks the validity of the sender's certificate;
- generates a unique identifier for each delivery (DeliveryId);
- decodes the message from base64;
- performs technical checks on the XML file:
  - checks the validity of the file name (as per the MMSR delivery file name convention);
  - checks the integrity of the XML file (namespace, UTF-8 encoding, etc.);
  - checks the validity of the XML headers (presence of appropriate Business Application Header and Document Header, existence of LEI, etc.);
- checks the uniqueness of the Business Application Header across all existing

deliveries on the basis of the sender and the BizMsgIdr;

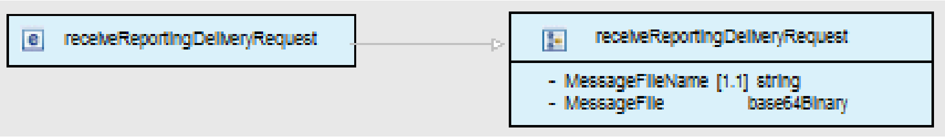
- checks the sender’s authorisation for the relevant reporting agents and market segments.

The ReceiveReportingDeliveryResponse is the response from the MMSR system. The response makes the DeliveryId immediately available to the sender.

7.2. ReceiveDeliveryService input parameters

The web service input parameters are listed below:

Input
ReceiveReportingDeliveryRequest


receiveReportingDeliveryRequest
receiveReportingDeliveryRequest
- MessageFileName [1.1] string
- MessageFile base64Binary

The request has two parameters:

MessageFileName (mandatory): file name of the submission. Please comply with the file name conventions described in Section 6 Technical requirements of the MMSR file*: MMSR delivery file name.*

MessageFile (mandatory): XML file encoded in base 64.

Here is an example of a request:

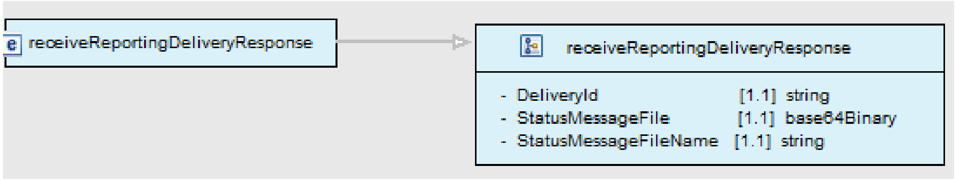
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">

   <soapenv:Body>

      <ecb:ReceiveReportingDeliveryRequest>


<ecb:MessageFileName>auth.012.001.01.7LTWFZYICNSX8D621K86.20170116.00
02</ecb:MessageFileName>

         <ecb:MessageFile>UEsDBBQACAgIACtUbEUAA...</ecb:MessageFile>
      </ecb:ReceiveReportingDeliveryRequest>

    </soapenv:Body>

  </soapenv:Envelope>
```

7.3. ReceiveDeliveryService output parameters

The web service output parameters are listed below:

Output

ReceiveReportingDeliveryResponse


e receiveReportingDeliveryResponse
receiveReportingDeliveryResponse
- DeliveryId [1.1] string
- StatusMessageFile [1.1] base64Binary
- StatusMessageFileName [1.1] string

The response has three parameters:

- **DeliveryId** (mandatory): The unique delivery identifier generated by the MMSR system. This identifier is necessary to track the submission in the MMSR system.

- **StatusMessageFile** (mandatory): ISO 20022 XML document containing the technical status message, including a list of XSD errors where applicable. Further details on the structure of the technical status message are given below in Section 10.4 Technical errors.

If the sender's certificate does not authorise it to submit a file to the MMSR system, the response from the web service will contain only a SOAP exception (with no DeliveryId, no StatusMessageFile and no StatusMessageFileName). See Section 10.4.2 SOAP_exceptions for a list of SOAP exceptions.

- **StatusMessageFileName** (mandatory): ISO 20022 XML document name corresponding to the ReportingStatusMessageFile parameter. The naming conventions are described in Section 7.5 MMSR status message file name.

Here is an example of a response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb=" http://eu.ecb.mmsr/">

  <soapenv:Body>

    <ecb:ReceiveReportingDeliveryResponse>

<ecb:DeliveryId>20150822-090622-100002-pr433a</ecb:DeliveryId>

      <ecb:StatusMessageFile>
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ…</ecb:StatusMessageFile>

<ecb:StatusMessageFileName>
auth.028.001.01.XXXXXXXXXXXXXXXXXX00.20150822-090622-100002-
pr433a</ecb:StatusMessageFileName>

      </ecb:ReceiveReportingDeliveryResponse>

  </soapenv:Body>

</soapenv:Envelope>
```

7.4. MMSR technical status message

The ECB technical status message is composed of three elements:

- a Business Application Header;
- a Reporting Header;
- a Reporting Message

The technical status message is an ISO 20022 compliant XML message that follows the schema (XSD) auth.028.001.01 (see Annex I: Technical documentation).

The technical status message is structured as follows:

a) A mandatory report status tag, (<RptSts>), which indicates the overall technical status of the delivery:

Data	XML tag	Format	Mandatory?	Description
Report status	<RptSts>	Text	Yes	Report technical status: ACTC, INCF, CRPT

The meaning of the technical status is as follows:

Code	Name	Meaning
ACTC	AcceptedTechnicalValidation	The file submitted does not contain technical errors.
INCF*	IncorrectFilename	The file submitted does not comply with the rules set out in *Section 6: MMSR delivery file name*.
CRPT*	CorruptedFile	The file submitted does not comply with the required XML schema, ISO 20022 elements, or other header requirements, see Section 5.3 *Details on the synchronous step of an MMSR submission*, paragraph 3 and 4.

* In the event of INCF or CRPT status, MMSR is not able to process the received file and its information cannot be extracted. However, the status message has some mandatory fields. These fields will be populated with default values (see Appendix II: XML examples).

b) An optional validation rule block (<VldtnRule>), which appears only when the report status is INCF or CRPT:

Data	XML tag	Format	Mandatory?	Description
Error code	<Id>	Text	No	Unique and unambiguous identification of a technical validation rule as per table below.
Description	<Desc>	Text	No	Description as per table below

Error code	Scope	Meaning
UTF8	File	Message is not UTF-8-encoded
XSD	File	Message is not ISO 20022-compliant
SEGMENT	File	Segment code is incorrect in MsgDefIdr
DIFFERENT_SEGMENT	File	Segment in reported document is not the segment indicated in MsgDefIdr
BUSINESS_SERVICE	File	Business service code is incorrect in BizSvc
RECEIVER_LEI	File	Receiver’s LEI is incorrect
NO_HABILITATION	File	Sender not authorised to report for this reporting agent on this segment
DUPLICATE_HEADER	All deliveries	AppHdr has the same BizMsgIdr and sender as a previous message

Below is an XML example of an ECB status message containing an error relating to the BusinessService XML tag:

```
<MMSRMessage>
  <Document>
     <MnyMktSttstclRptStsAdvc>
          <RptSts>CRPT</RptSts>
         <VldtnRule>
           <Id>BUSINESS_SERVICE</Id>
           <Desc>the business service code is incorrect in BizSvc</Desc>
        </VldtnRule>
     </MnyMktSttstclRptStsAdvc>
   </Document>
</MMSRMessage>
```

More examples can be found in Annex II: XML examples.

See also Annex I: Technical documentation

7.5. MMSR status message file name conventions

The names of MMSR status report messages follow the pattern below:

<MESSAGE DEFINITION IDENTIFIER>.<LEI>.<DELIVERY ID>

Variables	Description
<MESSAGE DEFINITION IDENTIFIER>	Length: 15 characters Message definition identifier. Value expected: *auth.028.001.01*
<LEI>	Length: 20 characters Reporting agent’s LEI
<DELIVERY ID>	Length: 29 characters Delivery Id generated by the MMSR ReceiveDeliveryService Format: [YYYY][MM][DAY: 1 to 31]-[HH][MM][SS]-[INCREMENTALNUMBER]-[MMSR INTERNAL VALUE]

For example, the first secured market status message made available to UniCredit Bank Austria AG on 16 January 2017 would have the following name:

auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.20170116-153227-0001-pr433a

Note that the same filename convention applies to the message produced by the *ReceiveDeliveryService* and to that produced by the *GetFeedbackService*.

7.6. WSDL integration

The ReceiveDeliveryService can be integrated using a WSDL file (ReceiveDeliveryService.wsdl).
The WSDL file can be found in Annex I: Technical documentation

7.7. Web service URL

Environment	**URL**
Production	https://mmsr-a2a.escb.eu/ws/MMSR_ReceiveDeliveryService_1_0_0_PRD
Pre-production	https://a-mmsr-a2a.escb.eu/ws/MMSR_ReceiveDeliveryService_1_0_0_HOMOL

8. “GetFeedbackService” web service

8.1. Introduction

Following the submission of the data file, the MMSR system will execute a set of business validation checks and produce a business status report message detailing any errors.

8.2. Description of the “GetFeedbackService” web service

The *GetFeedbackService* web service provides details of validation errors encountered during business validation checks.

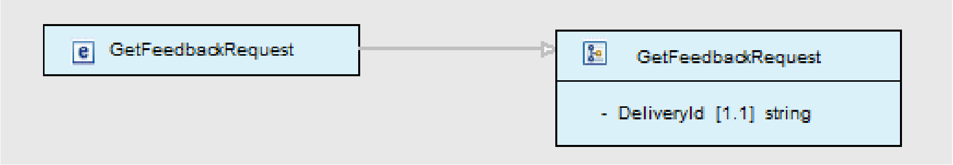
The sender can access this web service using the *DeliveryId* derived from the ReceiveReportingDeliveryResponse in order to obtain a business status message detailing the results of the business validation checks. This web service is described below.

8.3. “GetFeedbackService” input parameter

The web service input parameter is detailed below

Input

GetFeedbackRequest



The input parameter is defined as follows:

- **DeliveryId** (mandatory): The unique delivery identifier generated by the MMSR system and provided in the ReceiveDeliveryResponse. This is necessary to monitor the submission in the MMSR system.

Here is an example of such a request:

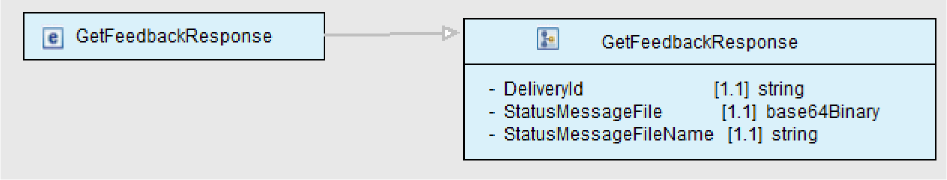
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">

  <soapenv:Body>

    <ecb:GetFeedbackRequest>

      <ecb:DeliveryId>20150822-090622-100002-pr433a</ecb:DeliveryId>

    </ecb:GetFeedbackRequest>

  </soapenv:Body>

</soapenv:Envelope>
```

8.4. “GetFeedbackService” output parameter and business status message

The web service output parameters are listed below:

Output

GetFeedbackResponse


GetFeedbackResponse
GetFeedbackResponse
- DeliveryId [1.1] string
- StatusMessageFile [1.1] base64Binary
- StatusMessageFileName [1.1] string

The response[3] has three parameters:

DeliveryId (mandatory): the unique delivery identifier.

The **StatusMessageFileName:** ISO 20022 XML document name corresponding to the ReportingStatusMessageFile parameter. This is a concatenation involving the DeliveryId and has the fixed label “StatusMessage”.

Here is an example of a StatusMessageFileName:

auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.20170101-153227-100055-pr433a

StatusMessageFile: ISO 20022 XML document containing:

- for submissions that resulted in an ACTC status code on the technical validation step described in Section 7.4 MMSR technical status message, the business status message for the submission
- for submissions that resulted in an INCF or a CRPT status code instead, the system cannot produce a business status message. In this case, the **StatusMessageFile** will be a copy of the technical status message instead (see Section 7.4 MMSR technical status message).

The *business status message* has the following structure and contents[4]:

a) A mandatory report status tag (<RptSts>) indicates the overall *business* status of the delivery:

	XML tag	Format	Mandatory?	Description
Report status	<RptSts>	Text	Yes	Report business status: ACTC, PART, RJCT

with the following possible meanings:

Code	Name	Meaning
ACPT	Accepted	The report has been accepted; no transactions have been rejected.
PART	PartiallyAccepted	One or more transactions in the report have been rejected and require re-submission with CORR status. Accepted transactions should not be resubmitted.
RJCT	Rejected	The report has been rejected due to errors in the report’s header. Transactions have not been processed at all so all of them must be resubmitted with the same status as in the rejected submission (see Section 5.4.Details on the asynchronous step of an MMSR submission)

In case the file is rejected (RJCT) due to errors in the report’s header (Section 11.2.5 Example of a StatusMessageFile in the event of a rejected file), the <Document> tag in the message will contain a list of the detected errors. The validation rules

[3] If the sender’s reporting agent is not authorised to submit to the MMSR system, the MMSR system will respond with a SOAP exception. If the DeliveryId is not found, or if the sender is not correct, the StatusMessageFile and StatusMessageFileName fields in the response will be empty.

[4] The business status message follows the schema (XSD) auth.028.001.01, which is shared for both technical and business status messages. See Annex I: Technical documentation.

applicable to the reporting header of the MMSR message can be found in Section 1 of the MMSR Data Quality Checks.

b) When the submission has overall business status ACPT or PART, a transaction status block (<TxSts>) will indicate the status of the reported transactions:

Code	Name	Meaning
ACPT	Accepted	The transaction has been accepted.
WARN	Warning	The transaction has been accepted, but with warnings.
RJCT	Rejected	The transaction has been rejected.

using the following XML structure:

Data	XML tag	Format	Mandatory?	Description
Unique transaction identifier	<UnqTxIdr>	Text	No	Unique transaction identifier
Proprietary transaction identifier	<PrtryTxId>	Text	Yes	Proprietary transaction identifier
Transaction status	<TrSts>	Text	Yes	Transaction status: ACPT, WARN or RJCT The transaction status is subject to the most stringent application of the validation rules. For example, a transaction has the status WARN if it contains two WARNs; a transaction has the status RJCT if it contains one WARN and one RJCT.
Validation rules	<VldtnRule>	XML block	No	This block will not appear if the transaction status is ACPT.

The validation rules applicable to the individual transactions within an MMSR message can be found in Sections 2 to 5 of the MMSR Data Quality Checks.

c) An optional validation rule block at transaction level (<VldtnRule>), when a transaction status (<TxSts>) is WARN or RJCT:

Data	XML tag	Format	Mandatory?	Description
Identification	<Id>	Text	No	Unique and unambiguous identification of a validation rule. This is derived from the Data Quality Checks spreadsheet.
Description	<Desc>	Text	No	Further information on the validation rule. This is the description field in the Data Quality Checks spreadsheet.
Issuer	<Issr>	Text	No	Value will be always “ECB MMSR”.

Below is an example of a response from the *GetFeedbackResponse* service:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">

    <soapenv:Body>

      <ecb:GetFeedbackResponse>

       <ecb:StatusMessageFile>
       PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ…</ecb:StatusMessageFile>

<ecb:StatusMessageFileName>auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.2017
0101-153227-100055-pr433a </ecb:StatusMessageFileName>

      </ecb:GetFeedbackResponse>

    </soapenv:Body>

 </soapenv:Envelope>
```

Below is an ISO 20022 XML example of a business status message containing two transactions with the status WARN:

```
<MMSRMessage>
   <Document>
        <MnyMktSttstclRptStsAdvc>

            <StsRptHdr>
                <RptgAgt>MMSRREPORTINGAGENT03<RptgAgt>
                <RptgPrd>
                    <FrDtTm>2015-06-09T17:25:41.000+02:00</FrDtTm>
                    <ToDtTm>2016-06-09T17:25:41.000+02:00</ToDtTm>
                </RptgPrd>
                <RptSts>ACPT</RptSts>
            </StsRptHdr>

            <TxSts>
                 <UnqTxIdr>c9f4c390-0ebb-11e5-857d-a0a8cd63461</UnqTxIdr>
                 <PrtryTxId>c9f4c660-0ebb-11e5-857d-a0a8cd63462</PrtryTxId>
                <Sts>WARN</Sts>
                <VldtnRule>
                    <Id>DQU1801</Id>
                    <Desc>DQU1801 - BASIS POINT SPREAD [BASIS POINT SPREAD
value] provided where TYPE OF RATE is fixed.</Desc>
                    <Issr>ECB MMSR</Issr>
                </VldtnRule>
                <VldtnRule>
                    <Id>DQU1701</Id>
                    <Desc>DQS1701 - REFERENCE RATE INDEX [REFERENCE RATE
INDEX] provided when TYPE OF RATE is FIXE.</Desc>
                    <Issr>ECB MMSR</Issr>
                </VldtnRule>
            </TxSts>

            <TxSts>
                 <UnqTxIdr>c9f4c390-0ebb-11e5-857d-a0a8cd63461</UnqTxIdr>
                 <PrtryTxId>c9f4c660-0ebb-11e5-857d-a0a8cd63462</PrtryTxId>
                <Sts>WARN</Sts>
                <VldtnRule>
                    <Id>DQU1801</Id>
                <Desc>DQU1801 - BASIS POINT SPREAD [BASIS POINT SPREAD
value] provided where TYPE OF RATE is fixed.</Desc>
                <Issr>ECB MMSR</Issr>
             </VldtnRule>
          </TxSts>

       </MnyMktSttstclRptStsAdvc>
    </Document>
</MMSRMessage>
```

8.5. WSDL integration

The "GetFeedbackService" web service can be integrated using a WSDL file (GetFeedbackService.wsdl). The WSDL file can be found in *Annex I: Technical documentation*

8.6. Web service URL

Environment	**Binding URL**
Production	https://mmsr-a2a.escb.eu/ws/MMSR_GetFeedbackService_1_0_0_PRD_
Pre-production	https://a-mmsr-a2a.escb.eu/ws/MMSR_GetFeedbackService_1_0_0_HOMOL

9. MMSR logging tool

9.1. Introduction

A dedicated web-based logging tool provides a UI allowing reporting agents to:

- review reception information;
- view and download dedicated human-readable reports;
- view and download status message reports;
- perform a manual upload of a file

9.2. E-mail notifications

The Transactional Module will send an e-mail notification to the RA in two instances:

1. It will send a reminder e-mail to the RA in the event of a missing submission (this is currently scheduled at 06:40 on every TARGET2 day).
2. E-mail alerts containing a list of rejected transactions awaiting correction will be sent to the RA one day prior to the expiry of the specified correction period.

10. Availability of the MMSR system

The MMSR system is available 24/7, for both A2A submissions and any interaction via the UI.

10.1. Connection modalities

MMSR uses an internet-based A2A channel. Senders are able to call the MMSR web services using an open SOAP standard. Web service calls are synchronous.

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.

The URL of the WSDL binding (<soap: address location>) must be changed according to the MMSR environment the sender is trying to reach:

Platform	Binding URL
Pre-production	https://a-mmsr-a2a.escb.eu/
Production	https://mmsr-a2a.escb.eu/

10.2. Contacting the MMSR support

MMSR technical support can be reached by email at: MMSR_MOA@banque-france.fr

10.3. Maintenance notifications

In case of maintenance, Banque de France sends an email to all RA technical contacts.

10.4. Technical errors

10.4.1. Status message technical errors

Error code	Scope	Meaning
UTF8	File	Message is not UTF-8-encoded
XSD	File	Message is not ISO 20022-compliant
SEGMENT	File	Segment code is incorrect in MsgDefIdr
DIFFERENT_SEGMENT	File	Segment in reported document is not the segment indicated in MsgDefIdr
BUSINESS_SERVICE	File	Business service code is incorrect in BizSvc
RECEIVER_LEI	File	Receiver's LEI is incorrect
NO_HABILITATION	File	Sender not authorised to report for this reporting agent on this segment
DUPLICATE_HEADER	All deliveries	AppHdr has the same BizMsgIdr and sender as a previous message

10.4.2. SOAP exceptions

Error code	Meaning	Type
TE_1	Not authorised	SOAP fault
TE_2	Not authorised	SOAP fault
TE_3	Not authorised	SOAP fault
TE_4	Not authorised	SOAP fault
TE_5	Not authorised	SOAP fault
TE_6	Not authorised	SOAP fault
TE_7	Not authorised	SOAP fault
TE_8	MMSR cannot read the file submitted	SOAP fault
TE_9	MMSR cannot determine delivery type	SOAP fault
TE_10	MMSR cannot transform delivery	SOAP fault
TE_11	Missing XSD file	SOAP fault
TE_12	Exception during XSD validation	SOAP fault
TE_13	Cannot instantiate a calendar	SOAP fault
TE_14	Cannot create StatusMessageFile	SOAP fault
TE_15	Cannot instantiate an XMLGregorianCalendar	SOAP fault
TE_16	ApplCorID is null or empty	SOAP fault
TE_17	ProcessGroup is null or empty	SOAP fault
TE_18	ProcessName is null or empty	SOAP fault
TE_19	Sender is null or empty	SOAP fault
TE_20	Cannot verify user's authorisations	SOAP fault
TE_21	Not authorised	SOAP fault

In the event of a SOAP fault, the web service response will contain only a SOAP exception (i.e. the ReceiveReportingDeliveryResponse will not contain the expected DeliveryId, MessageFile or MessageFileName). The sender will have to contact the MMSR Helpdesk or make a new submission.

11. Annexes

11.1. Annex I: Technical documentation

WSDL

The Web Services Description Language of the two webservices are available at the following URL: https://www.ecb.europa.eu/stats/money/mmss/shared/files/MMSR-Web_service_description-WSDL.zip

The zip file contains:

File name	Path in the zip	Related section	Function
ReceiveDeliveryService.wsdl	MMSR-WS-INTERNET\ReceiveDeliveryService.wsdl	Section 7	XML that describes tags and attributes that can be used in order to communicate with the ReceiveDeliveryService webservice
GetFeedbackService.wsdl	MMSR-WS-INTERNET\ GetFeedbackService.wsdl	Section 8	XML that describes tags and attributes that can be used in order to communicate with the GetFeedbackService webservice

TECHNICAL WRAPPER, ISO 20022 XSD and Message Definition Report

XML files attached to calls at the ReceiveDeliveryService webservice should be compliant with XML Schema Definitions available at the following URL: https://www.ecb.europa.eu/stats/money/mmss/shared/files/MMSR_XML_Schemas.zip

The zip file contains:

File name	Path in the zip	Related section	Function
MMSR_validation_ReportingMessages.xsd	MMSR - XML Schemas\Technical Wrapper\MMSR_validation_ReportingMessages.xsd	Section 7.2	Technical Message main wrapper that describes how MMSR report should be constructed (with a Header and Document part)
ECB_MMSR_BAH_head_001_ForReportingMessages_Schema_20150903.xsd	MMSR - XML Schemas\BAH_ForReportingMessages\ ECB_MMSR_BAH_head_001_ForReportingMessages_Schema_20150903.xsd	Section 7.2	ISO 20022 BAH XSD that describes Header (part common to all segments) for MMSR reports
auth.012.001.02.xsd	MMSR - XML Schemas\Schema_Doc\auth.012.001.02.xsd	Section 7.2	ISO 20022 XSD that describes Document part of the report for MMSR messages related to SECURED segment
auth.013.001.02.xsd	MMSR - XML Schemas\Schema_Doc\auth.013.001.02.xsd	Section 7.2	ISO 20022 XSD that describes Document part of the report for MMSR messages related to UNSECURED segment
auth.014.001.02.xsd	MMSR - XML Schemas\Schema_Doc\auth.014.001.02.xsd	Section 7.2	ISO 20022 XSD that describes Document part of the report for MMSR messages related to FX Swaps segment
auth.015.001.02.xsd	MMSR - XML Schemas\Schema_Doc\auth.015.001.02.xsd	Section 7.2	ISO 20022 XSD that describes Document part of the report for OIS messages related to

			SECURED segment
auth.028.001.02.xsd	MMSR - XML Schemas\Schema_Doc\auth.028.001.02.xsd	Section 8.4	ISO 20022 XSD that describes Document part of the status report for MMSR messages that is returned by the ReceiveDeliveryService webservice
ECB_MMSR_BAH_head_001_ForReportingMessages_Compact_20150903.pdf	MMSR - XML Schemas\BAH_ForReportingMessages\ECB_MMSR_BAH_head_001_ForReportingMessages_Compact_20150903.pdf	Section 7	Business Application Header (BAH) for MMSR messages
ECB_MMSR_BAH_head_001_ForStatusMessage_Compact_20150903.pdf	MMSR - XML Schemas\BAH_ForStatusMessages\ECB_MMSR_BAH_head_001_ForStatusMessage_Compact_20150903.pdf	Section 8	Business Application Header (BAH) for MMSR status reports
ISO20022_MDRPart2_MMSR_Maintenance_2017_v1.docx.pdf	MMSR - XML Schemas\ ISO20022_MDRPart2_MMSR_Maintenance_2017_v1.docx.pdf	Section 7	This document provides details of the Message Definitions for Money Market Statistical Reporting.

11.2. Annex II: XML examples

11.2.1. Example of a ReceiveReportingDeliveryRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soapenv:Body>
    <ecb:ReceiveReportingDeliveryRequest>

      <ecb:MessageFileName>auth.013.001.02.9W4ONDYI7MRRJYXY8R34.20150
804.0001</ecb:MessageFileName>

      <ecb:MessageFile>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ
</ecb:MessageFile>
    </ecb:ReceiveReportingDeliveryRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

11.2.2. Example of a ReceiveReportingDeliveryResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soap:Body>
    <ecb:ReceiveReportingDeliveryResponse>
      <ecb:DeliveryId>20150822-090622-100002-pr433a</ecb:DeliveryId>
     <ecb:StatusMessageFile>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ…</ecb:Sta
tusMessageFile>
     <ecb:StatusMessageFileName>auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.
20170116-090622-100002-pr433a</ecb:StatusMessageFileName>
    </ecb:ReceiveReportingDeliveryResponse>
  </soap:Body>
</soap:Envelope>
```

11.2.3. Example of a ReceiveReportingDeliveryResponse in the event of a corrupted file

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soap:Body>
    <ecb:ReceiveReportingDeliveryResponse>
      <ecb:DeliveryId>2015082-090622-100002-pr433a</ecb:DeliveryId>
      <ecb:StatusMessageFile>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ…
     </ecb:StatusMessageFile>
     <ecb:StatusMessageFileName>auth.028.001.01.7LTWFZYICNSX8D621K86.2
017016-090622-100002-pr433a </ecb:StatusMessageFileName>
    </ecb:ReceiveReportingDeliveryResponse>
  </soap:Body>
</soap:Envelope>
```

11.2.4. Example of a StatusMessageFile in the event of a corrupted file

In the event of a corrupt file name or a corrupted file, the MMSR system will not be able to process the submitted file or extract information.

Some fields will be filled with default values. In the example below, green characters denote default values and purple characters denote other values.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MMSRMessage
Xmlns:h="iso:std:iso:20022:tech:xsd:head.001.001.01"
xmlns:s="urn:iso:std:iso:20022:tech:xsd: auth.028.001.01"
xmlns="urn:iso:std:iso:20022:tech:xsd:head.003.001.01">
   <h:AppHdr>
      <h:Fr>
         <h:OrgId>
            <h:Id>
               <h:OrgId>
                  <h:Othr>
                     <h:Id> XXXXXXXXXXXXXXXXXXXX00</h:Id>
                     <h:SchmeNm>
                        <h:Cd>LEI</h:Cd>
                     </h:SchmeNm>
                  </h:Othr>
               </h:OrgId>
            </h:Id>
         </h:OrgId>
      </h:Fr>
      <h:To>
```

```
    <h:OrgId>
        <h:Id>
            <h:OrgId>
                <h:Othr>
                    <h:Id> XXXXXXXXXXXXXXXXXXXX00</h:Id>
                    <h:SchmeNm>
                        <h:Cd>LEI</h:Cd>
                    </h:SchmeNm>
                </h:Othr>
            </h:OrgId>
        </h:Id>
    </h:OrgId>
</h:To>
<h:BizMsgIdr>IREF012345</h:BizMsgIdr>
<h:MsgDefIdr> auth.028.001.01</h:MsgDefIdr>
<h:BizSvc>ECB_MMSR_TEST</h:BizSvc>
<h:CreDt>9999-12-31T00:00:00.0Z</h:CreDt>
<h:Rltd>
<h:Fr>
    <h:OrgId>
        <h:Id>
            <h:OrgId>
                <h:Othr>
                    <h:Id> XXXXXXXXXXXXXXXXXXXX00</h:Id>
                    <h:SchmeNm>
                        <h:Cd>LEI</h:Cd>
                    </h:SchmeNm>
                </h:Othr>
            </h:OrgId>
        </h:Id>
    </h:OrgId>
</h:Fr>
<h:To>
    <h:OrgId>
        <h:Id>
            <h:OrgId>
                <h:Othr>
                    <h:Id> XXXXXXXXXXXXXXXXXXXX00</h:Id>
                    <h:SchmeNm>
                        <h:Cd>LEI</h:Cd>
                    </h:SchmeNm>
                </h:Othr>
            </h:OrgId>
        </h:Id>
    </h:OrgId>
</h:To>
```

```
  </h:Rltd>
  </h:AppHdr>
  <s:Document>
    <s:MnyMktSttstclRptStsAdvc>
      <s:RptHdr>
        <s:RptgAgt>XXXXXXXXXXXXXXXXXXXX00</s:RptgAgt>
        <s:RefPrd>
          <s:FrDtTm>9999-12-31T00:00:00.0Z</s:FrDtTm>
          <s:ToDtTm>9999-12-31T00:00:00.0Z</s:ToDtTm>
        </s:RefPrd>
      </s:RptHdr>
      <s:ScrdMktRpt>
        <s:Tx>
          <s:RptdTxSts>CRPT</s:RptdTxSts>
         <s:VldtnRule>
         <s:Id>XSD </s:Id>
          <s:Desc>[Error – line : 42 – Column : 49] : cvc-type.3.1.3 : The value
‘MMSRREPORTINGAGENT0’ of element ‘s:RptgAgt’ is not valid.</s:Desc>
         </s:VldtnRule>
    </s:Tx>
    </s:ScrdMktRpt>
    </s:MnyMktSttstclRptStsAdvc>
  </s:Document>
</MMSRMessage>
```

11.2.5. Example of a StatusMessageFile in the event of a rejected file

In the event of a rejected file, the MMSR system is able to process the submitted file and extract its information but due to errors on headers fields the report is not integrated.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:MMSRMessage xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01"
xmlns:ns2="urn:iso:std:iso:20022:tech:xsd:auth.028.001.01"
xmlns:ns3="urn:iso:std:iso:20022:tech:xsd:head.003.001.01">
    <AppHdr>
        <Fr>
            <OrgId>
                <Id>
                    <OrgId>
                        <Othr>
                            <Id>549300DTUYXVMJXZNY75</Id>
                            <SchmeNm>
<Cd>LEI</Cd>
                            </SchmeNm>
                        </Othr>
                    </OrgId>
                </Id>
            </OrgId>
        </Fr>
        <To>
            <OrgId>
                <Id>
                    <OrgId>
                        <Othr>
                            <Id>LEIOFREPORTINGAGENT</Id>
                            <SchmeNm>
<Cd>LEI</Cd>
                            </SchmeNm>
                        </Othr>
                    </OrgId>
                </Id>
            </OrgId>
        </To>
```

```
<BizMsgIdr>20230923-020008-107336-0800sv</BizMsgIdr>
    <MsgDefIdr>auth.028.001.01</MsgDefIdr>
    <BizSvc>ECB_MMSR_PROD</BizSvc>
    <CreDt>2023-09-23T00:00:09.023Z</CreDt>
    <Rltd>
      <Fr>
        <OrgId>
          <Id>
            <OrgId>
              <Othr>
<Id> LEIOFREPORTINGAGENT</Id>
<SchmeNm>
  <Cd>LEI</Cd>
</SchmeNm>
              </Othr>
            </OrgId>
          </Id>
        </OrgId>
      </Fr>
      <To>
        <OrgId>
          <Id>
            <OrgId>
              <Othr>
<Id>549300DTUYXVMJXZNY75</Id>
<SchmeNm>
  <Cd>LEI</Cd>
</SchmeNm>
              </Othr>
            </OrgId>
          </Id>
        </OrgId>
      </To>
      <BizMsgIdr>20230923-020008-106165-0802sv-RPT</BizMsgIdr>
      <MsgDefIdr>auth.015.001.02</MsgDefIdr>
      <BizSvc>ECB_MMSR_PROD</BizSvc>
      <CreDt>2023-09-23T00:00:03.000Z</CreDt>
    </Rltd>
  </AppHdr>
```

```
    <ns2:Document>
    <ns2:MnyMktSttstclRptStsAdvc>
      <ns2:StsRptHdr>
        <ns2:RptgAgt>XXXXXXXXXXXXXXXXXXXXX</ns2:RptgAgt>
        <ns2:RptgPrd>
          <ns2:FrDtTm>2023-09-22T18:00:00.000Z</ns2:FrDtTm>
          <ns2:ToDtTm>2023-09-25T00:00:00.000Z</ns2:ToDtTm>
        </ns2:RptgPrd>
        <ns2:RptSts>RJCT</ns2:RptSts>
        <ns2:VldtnRule>
          <ns2:Id>DQH602</ns2:Id>
          <ns2:Desc>DQH602 - Creation Date [2023-09-23T00:00:03Z] is
before Reference Period end [2023-09-25T00:00:00Z].</ns2:Desc>
          <ns2:Issr>ECB_MMSR</ns2:Issr>
        </ns2:VldtnRule>
      </ns2:StsRptHdr>
    </ns2:MnyMktSttstclRptStsAdvc>
  </ns2:Document>
</ns3:MMSRMessage>
```

11.2.6. Example of a GetFeedbackRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soapenv:Header/>
  <soapenv:Body>
    <ecb:GetFeedbackRequest>
      <ecb:DeliveryId>20150822-090622-100002-pr433a</ecb:DeliveryId>
    </ecb:GetFeedbackRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

11.2.7. Example of a GetFeedbackResponse

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soapenv:Header/>
```

```
   <soapenv:Body>
      <ecb:GetFeedbackResponse>
       <ecb:StatusMessageFile>cid:10335101
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZ…28457</ecb:StatusMessageFile>
<ecb:StatusMessageFileName>auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.2017
0101-153227-100055-pr433a </ecb:StatusMessageFileName>
      </ecb:GetFeedbackResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

11.3. Annex III: “ReceiveDeliveryService” validation rules

Refer to the *Money Market Statistical Reporting (MMSR)* – Data Quality Checks document.