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

Table of contents

1 Glossary

Term	Definition
MMSR system	The 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.
Submission	File submission relates to the reception in the Transactional Module of a dataset sent by a sender.
A2A	Application-to-application submission process
U2A	User-to-application submission process
Sender	The sending application
Receiver	MMSR Transactional Module
ESCB	The European System of Central Banks (ESCB) is composed of the European Central Bank (ECB) and the national central banks (NCBs) of all 28 European Union (EU) Member States.
IAM	IAM is a shared ESCB service used to authenticate senders and manage their access rights for the Transactional Module.
RA	Reporting agents (RAs) are commercial banks that report to the MMSR system. A commercial bank may be the sender of the submission, or it may delegate that to a third party.
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.
CAF	In the European context, CAF relates to the multi-acceptance of certificates which are compatible with the MMSR system.

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 Single Supervisory Mechanism (SSM) in accordance with Council Regulation (EU) No 1024/2013.

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 Governing Council of the ECB.

3 Scope of document

This document describes the submission and feedback processes to be followed by reporting agents reporting directly to the MMSR Transactional Module.

The submission and feedback processes to be followed by the NCBs are described in a dedicated document. The NCBs, which implement local MMSR collection platforms, are responsible for instructing the reporting agents in their jurisdiction.

4 Description of MMSR processes

The ECB will collect daily transactional data on secured, unsecured and derivative transactions. The data will be reported in a unified Extensible Markup Language (XML) format (ISO 20022-compliant) on a daily basis (especially in the first wave of reporting by 53 MFIs – i.e. as of 1 April 2016). Some weekly and monthly data may also be reported (in the second wave of reporting – i.e. as of January 2017).

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

Data files will undergo validation checks when they are received by the MMSR system, and automated status messages will be sent back to the sender.

The validation checks will involve a threshold based on the percentage of erroneous transactions across a single data file. Where that percentage is above the specified threshold, the status message containing the results of the validation checks will also indicate that the file has been rejected, requesting the resubmission of the dataset.

Reporting agents will be able to monitor the data files and the status messages containing the results of the validation checks, as well as reports via 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 are able to make web service calls using an SOAP open standard.

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

4.1 Using MMSR

The standard use of MMSR is as follows:

1. RA prepares the RA delivery message
   Please refer to the Reporting Instructions document.

2. RA submits the RA delivery message to MMSR
   Please refer to *Section 4.3: Synchronous submission part of an MMSR message*.
   Please refer to *Section 6: "ReceiveDeliveryService"*.

3. RA simultaneously receives the DeliveryId and a first technical status message. This status message only contains the results of technical checks and indicates whether the NCB message is ISO 20022-compliant.
   Please refer to *Section 6.2.2: Web service output parameters*.

4. Around ten minutes later, the RA can use the GetFeedbackService in order to obtain the second functional status message containing the results of the validation checks.
   Please refer to *Section 7:"GetFeedbackService" web service*.

Please note that the GetFeedbackService can be used several times (e.g. in order to receive the second functional status message again).

4.2 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 send MMSR an RA delivery message with the DataSetAction XML tag populated with "NOTX".

- The BusinessService XML tag in the Business Application Header can be populated with one of the following two values:

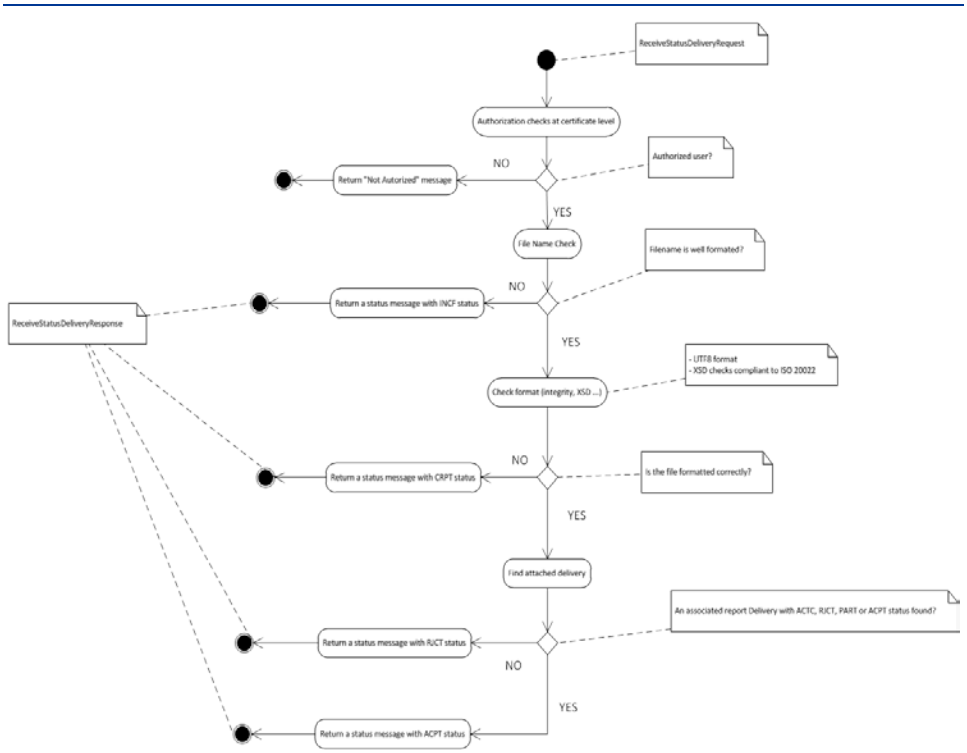
  (a) "ECB_MMSR_PROD" if the RA wants to send a delivery (standard case);

  (b) "ECB_MMSR_TEST" if the RA wants to test the technical channel. In that case, the MMSR delivery process (described in *Section 4.3: Synchronous submission part of an MMSR message*) will apply in full. Hence, MMSR will send the RA a technical status message. Nevertheless, no data will be stored in MMSR (including the DeliveryId, which cannot be used to call the GetFeedbackService).

Please also refer to *Annex III: Message Definition Report*.

4.3 Synchronous submission part of an MMSR message

The synchronous submission part of an MMSR message from a reporting agent to the ECB can be visualised using the following flow chart:



Details of that process are provided below. (See also *Section 6:"ReceiveDeliveryService"*.)

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 *Section 6.3.5: MMSR delivery file name*), a status message (auth.028.001.01) is returned with a status of "INCF". In this case, the RA field of the StatusMessageFile is populated with the "unknown LEI" XXXXXXXXXXXXXXXXXX00 and the reporting period field is populated with the "unknown date" 9999-12-31.

(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 status message (auth.028.001.01) is returned with a status of “CRPT”. In this case, the RA field of the StatusMessageFile is populated with the “unknown LEI” XXXXXXXXXXXXXXXXX00 and the reporting period field is populated with the “unknown date” 9999-12-31. Consequently, the RA is required to resubmit the relevant transactions with the same status as the previous submission.

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

4. Checks on file information:

   (a) If the MessageDefinitionIdentifier provided is different from the content of the report, or if the receiver’s legal entity identifier (LEI) is not the ECB’s, a status message (auth.028.001.01) is returned with a status of “CRPT”.

   (b) If the sender is not authorised to deliver for a particular segment, a status message (auth.028.001.01) is returned with a status of “CRPT”.

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

5. Quality check (asynchronous) process

6. Status message (auth.028.001.01) returned with a status of “ACTC”

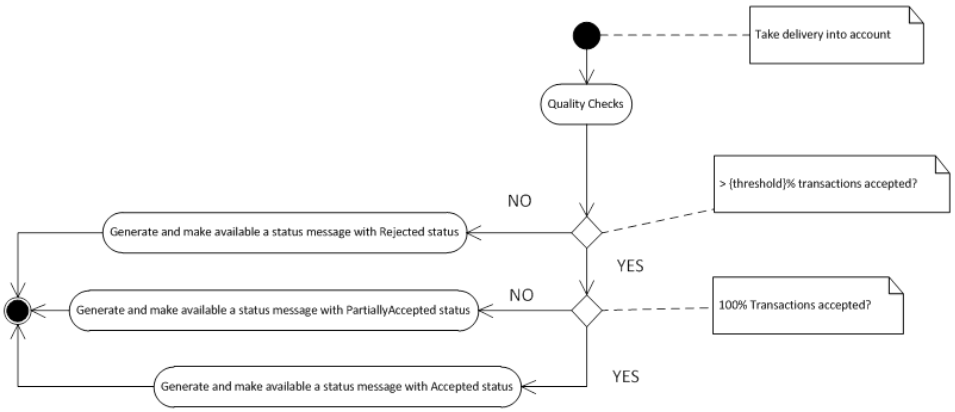
The valid reported transaction statuses are defined below:

Code	Name	Meaning
AMND	Amendment	Transaction amends a previously sent transaction
CANC	Cancellation	Transaction requests the deletion/cancellation of a previously sent transaction
CORR	Correction	Transaction corrects errors in a previously sent transaction
NEWT	NewTransaction	Transaction is a new transaction

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.

4.4 Asynchronous submission part of an MMSR message

The asynchronous submission part of an MMSR message from a reporting agent to the ECB can be visualised using the following flow chart:



Details of that process are provided below.

1. A correctly formatted file that complies with the file name conventions is received by the system.

2. Quality checks involve a threshold based on the percentage of erroneous transactions across a single data file. If that percentage is above the specified threshold, the status message containing the results of the validation checks will also indicate that the file has been rejected, requesting the resubmission of the dataset. For details of the rules governing those checks, see the *Money Market Statistical Reporting (MMSR) – Data Quality Checks* document.

   (a) If the percentage of erroneous transactions in the file is above the threshold level (i.e. more than 80% are rejected transactions), the Transactional Module will generate a status message with a status of "**RJCT**" (rejected). This status message containing the results of the quality checks is available via the feedback service (see *Section 7: "GetFeedbackService" web service*). In this case, the reporting agent must resubmit all transactions, using the same status as in the previous submission.

   (b) If the file contains erroneous transactions, but that percentage is below the threshold, the Transactional Module will generate a status message with a status of "**PART**" (partially accepted). This status message containing the results of the quality checks is available via the feedback service (see *Section 7: "GetFeedbackService" web service*). In this case, the reporting agent must resubmit only the rejected transactions, using the transaction status "CORR".

(c) If the file does not contain any erroneous transactions, the Transactional Module will generate a status message with a status of "**ACPT**" (accepted). This status message containing the results of the quality checks is available via the feedback service (see *Section 7: "GetFeedbackService" web service*).

4.5 MMSR web services

The MMSR system offers two web services:

- The submission process (described in Section 6: "ReceiveDeliveryService") is composed of two parts:
  - The sender transmits ISO 20022 XML files to MMSR. XML data are transformed and loaded into the Transactional Module. The XML files undergo a set of technical checks (checking for corrupted files, checking authorisation, ensuring that file names comply with naming conventions, etc.).
  - The sender immediately receives an MMSR response containing a unique identifier (which allows it to track the submission) and a status message containing a list of technical errors.
- The second web service (described in *Section 7*: *"GetFeedbackService" web service*) allows a sender to obtain more details about its submission. The sender can use this web service to obtain a list of validation errors by citing the identifier obtained at the end of the submission process. This can only be requested after the submission process has been completed.

This document provides technical details on the use of these two web services.

5 Prerequisites for the use of web services

MMSR is protected by certificate authentication. Hence, the sender must obtain the following:

- two IAM user accounts:
  - an IAM production user account;
  - an IAM pre-production user account;
- a valid ESCB software certificate (not expired or revoked) delivered by a CAF-compliant PKI. This certificate will be mapped by the MMSR Helpdesk to both IAM user accounts.

Important: MMSR will be available in pre-production as of 4 January 2016 and production as of 1 April 2016.

The MMSR Helpdesk will assist with the obtaining of IAM accounts and the ESCB software certificate.

For its part, the sender will have to:

- install the certificate on the sending server;
- call the MMSR web services.

6 "ReceiveDeliveryService" web service

6.1 Introduction

As of 1 April 2016, reporting agents must submit ISO 20022 XML files to the MMSR system, as detailed in the Reporting Instructions.

The MMSR system provides a web service called the "ReceiveDeliveryService" for this purpose.

This section describes that web service and provides a number of examples to aid your understanding.

6.2 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 base 64;
- performs technical checks on the XML file:
    - checks the validity of the file name (as described in Section 6.3.5.5: MMSR delivery file name);
    - 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.

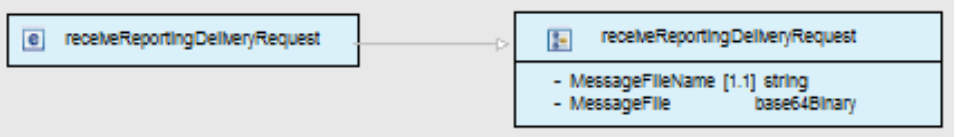
The ReceiveReportingDeliveryResponse is the response from the MMSR system. The system immediately sends the DeliveryId to the sender.

6.2.1 Web service input parameters

The web service input parameters are listed below:

Input

ReceiveReportingDeliveryRequest



The request has two parameters:

- **MessageFileName** (mandatory): file name of the submission. Please comply with the file name conventions described in *Section 6.3.5: 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>
```

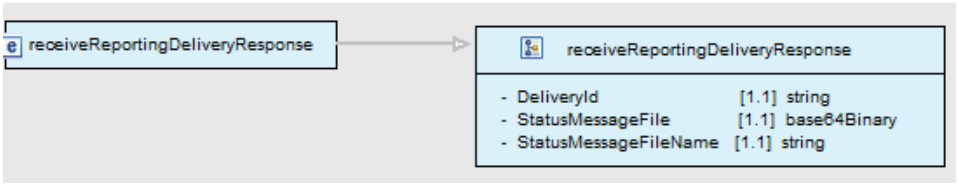
```
    </ecb:ReceiveReportingDeliveryRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

6.2.2 Web service output parameters

The web service output parameters are listed below:

Output

ReceiveReportingDeliveryResponse



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 a list of XSD errors. Please note that only technical errors relating to the fields and values contained in the <AppHdr> XML tag are reported.

Please find below a list of error codes and their meanings:

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

For an exhaustive list of the various checks, see the technical rules on the ReceiveDeliveryService in *Annex III: Message Definition Report*.

The StatusMessageFile also indicates the status of the delivery:

Code	Name	Meaning
ACTC	AcceptedTechnicalValidation	The file submitted does not contain errors.
INCF1	IncorrectFilename	The file submitted does not comply with the rules set out in *Section 6.3.5: MMSR delivery file name* .
CRPT2	CorruptedFile	The file submitted does not comply with ISO 20022 XSD as defined in the Reporting Instructions, or some reported information is inconsistent. (For an exhaustive list of the various checks, see the technical rules on the ReceiveDeliveryService in *Annex III: Message Definition Report*.)

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 an SOAP exception (with no DeliveryId, no StatusMessageFile and no StatusMessageFileName). See *Section 10.4.1: Status message technical errors* for more information.

- **StatusMessageFileName** (mandatory): ISO 20022 XML document name corresponding to the ReportingStatusMessageFile parameter. The naming conventions are described in *Section 6.3.6: 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>
```

1 In the event of an incorrect file name, MMSR is not able to process the received file and we cannot extract information. However, the status message has some mandatory fields. These fields will be populated with default values (see Appendix II: XML examples).

2 In the event of a corrupted file, MMSR is not able to process the received file and we cannot extract information. However, the status message has some mandatory fields. Some of these fields will be populated with default values (see Appendix II: XML examples).

6.3 Message file structure conventions

6.3.1 Conceptual structure of an MMSR delivery message

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:

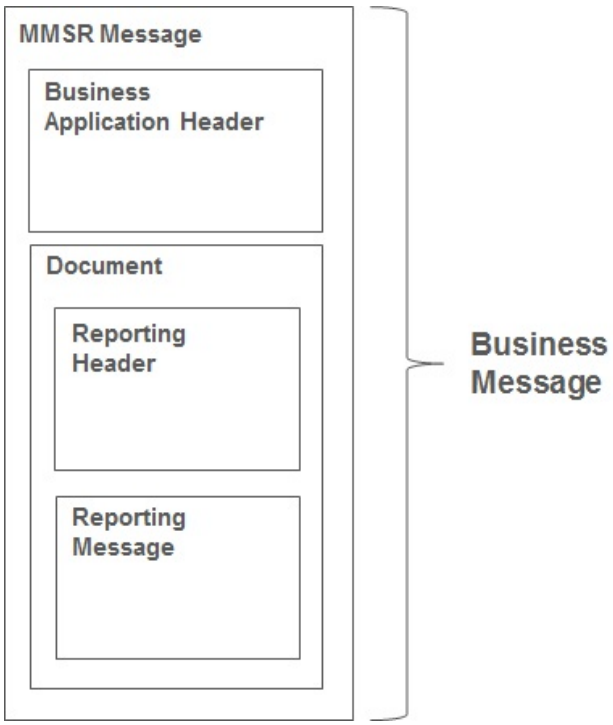
(a) a Business Application Header (BAH), which is used to identify the message and includes routing information;

(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.



6.3.2 MMSR status message

Like the delivery message, the ECB status message is composed of three elements:

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

The Reporting Message is structured as follows:

A mandatory report status tag (<RptSts>):

Data	XML tag	Format	Mandatory?	Description
Report status	<RptSts>	Text	Yes	Report status: INCF, CRPT, ACTC, RJCT, PART or ACPT See *Section 4.3: Synchronous submission part of an MMSR message.*

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

Data	XML tag	Format	Mandatory?	Description
Identification	<Id>	Text	No	Unique and unambiguous identification of a technical validation rule. This is derived from: the Data Quality Checks spreadsheet; *Section 10.4: Technical errors*.
Description	<Desc>	Text	No	Further information on the validation rule (only in the event of a technical error)

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>
```

An optional transaction status block (<TxSts>), which appears if the file is ACTC (accepted from a technical perspective) in the first technical phase:

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.

An optional validation rule block (<VldtnRule>), which appears only if the 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 XML example of an ECB 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>
```

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

See also *Annex III: Message Definition Report*

6.3.3 Technical wrapper

For the submission of the ISO 20022-compliant MMSR messages, a “technical wrapper” is used to structure 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: XML schemas.

6.3.4 Rules on sequencing of XML tags

Please note that 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.

6.3.5 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

6.3.6 MMSR status message file name conventions

The names of MMSR status report messages must 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 DeliveryId 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 sent to UniCredit Bank Austria AG on 16 January 2017 will have the following name:

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

6.4 WSDL integration

The ReceiveDeliveryService can be integrated using a WSDL file (ReceiveDeliveryService.wsdl).

The WSDL file can be found in *Annexes*

Annex I: XML schemas.

7 “GetFeedbackService” web service

7.1 Introduction

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

The validation checks will involve a threshold based on the percentage of erroneous transactions across a single data file. Where that percentage is above the specified threshold, the status message containing the results of the validation checks will also indicate that the file has been rejected, requesting the resubmission of the dataset.

7.2 Description of the “GetFeedbackService” web service

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

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

7.2.1 Web service 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 ReceiveReportingDeliveryResponse. This is necessary to monitor the submission in the MMSR system.

Here is an example of such a request:

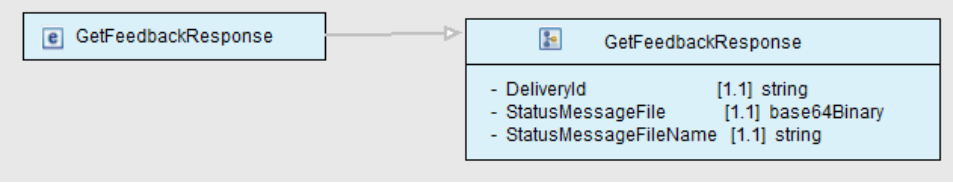
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soapenv:Body>
    <ecb:GetFeedbackRequest>
      <ecb:DeliveryId>20150822-090622-100002-pr433a</ecb:DeliveryId>
    </ecb:GetFeedbackRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.2 Web service output parameters

The web service output parameters are listed below:

Output

GetFeedbackResponse



The response has three parameters:

- **DeliveryId** (mandatory): the unique delivery identifier.
- **StatusMessageFile**:[3] ISO 20022 XML document containing the following:
  - technical errors relating to the fields and values contained in the <AppHdr> tag;
  - validation errors relating to the fields and values contained in the <Document> tag in the XML file.

For details of the validation rules, see *Annex IV: "ReceiveDeliveryService" validation*

The StatusMessageFile also indicates the status of the delivery:

Code	Name	Meaning
INCF	IncorrectFilename	The file submitted does not comply with the rules set out in *Section 6.3.5: MMSR delivery file name* .
CRPT	CorruptedFile	The file submitted does not comply with ISO 20022 XSD as defined in the Reporting Instructions, or some reported information is inconsistent. (For an exhaustive list of the various checks, see the technical rules on the ReceiveDeliveryService in *Annex III: Message Definition Report*.)
ACTC	AcceptedTechnicalValidation	The file submitted does not contain errors.
ACPT	Accepted	The report has been accepted; no transactions have been rejected.
PART	PartiallyAccepted	The report has been partially accepted. Some transactions have been accepted, but others have not yet been accepted.
RJCT	Rejected	The report has been rejected, as the number of rejected transactions exceeds the permitted threshold.

The StatusMessageFile also indicates the status of each transaction:

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.

If the sender's reporting agent is not authorised to submit to the MMSR system, the MMSR system will respond with an SOAP exception.

- **StatusMessageFileName**:[4] 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

Here is an example of a response:

[3] 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] If the DeliveryId is not found, or if the sender is not correct, the StatusMessageFile and StatusMessageFileName fields in the response will be empty.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ecb="http://eu.ecb.mmsr/">
  <soapenv:Body>
    <ecb:GetFeedbackResponse>
     <ecb:StatusMessageFile>cid:1033510128457</ecb:StatusMessageFile>
<ecb:StatusMessageFileName>auth.028.001.01.D1HEB8VEU6D9M8ZUXG17.20170101-153227-100055-pr433a </ecb:StatusMessageFileName>
    </ecb:GetFeedbackResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7.3 WSDL integration

The “GetFeedbackService” web service can be integrated using a WSDL file (GetFeedbackService.wsdl). The WSDL file can be found in *Annexes*

Annex I: XML schemas.

8 MMSR logging tool

8.1 Introduction

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

- review reception information;
- view and download dedicated human-readable reports;
- view and download status message reports.

More detailed information regarding the interface and the various reports will be provided at a later stage.

9 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.

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

10.1 Connection modalities

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

When an error occurs, an 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.

Please note that 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/

Acceptance tests are possible as of 4 January 2016.

10.2 Contacting the MMSR Helpdesk

Contact details for the pre-production and production phases will be distributed in a separate document.

10.3 Maintenance notifications

This information will be provided at a later stage.

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 an SOAP fault, the web service response will contain only an 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: XML schemas

WSDL

ReceiveDeliveryService wrapper	 ReceiveDeliveryService.wsdl
GetFeedbackService wrapper	 GetFeedbackService.wsdl

TECHNICAL WRAPPER

Technical wrapper	 ReportingMessages.xsd

ISO 20022 BAH Message XSD

MMSR – XML Schemas – BAH	 MMSR - XML Schemas - BAH.zip

ISO 20022 Messages XSD

MoneyMarketSecuredMarketStatisticalReportV01	 auth.012.001.02.xsd
MoneyMarketUnsecuredMarketStatisticalReportV01	 auth.013.001.02.xsd
MoneyMarketForeignExchangeSwapsStatisticalReportV01	 auth.014.001.02.xsd

MoneyMarketOvernightIndexSwapsStatisticalReportV01	auth.015.001.02.xsd
MoneyMarketStatisticalReportStatusAdviceV01	auth.028.001.01.xsd

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, we 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> XXXXXXXXXXXXXXXXXXX00</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>XXXXXXXXXXXXXXXXXXX00</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 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>
```

11.2.6 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: Message Definition Report

BAH for Reporting Message	 ECB_MMSR_BAH_he ad_001_ForReportinç
Reporting Message	 ISO20022_MDRPart2 _MMSR_Maintenance
Status Message	 ECB_MMSR_BAH_he ad_001_ForStatusMe

11.4 Annex IV: “ReceiveDeliveryService” validation rules

Please refer to the *Money Market Statistical Reporting (MMSR) – Data Quality Checks* document circulated together with the Reporting Instructions.