

REMIT

Agency's REMIT Information System Data Validation

Version 4.2

19 February 2018

Version history

| Version | Effective Date |
|---|-------------------|
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 1.0 | 9 March 2015 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 1.1 | 11 June 2015 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 2.0 | 28 September 2015 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 2.1 | 10 February 2016 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 3.0 | 17 March 2016 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 4.0 | 3 October 2016 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 4.1 | 15 March 2017 |
| AGENCY'S REMIT INFORMATION SYSTEM DATA VALIDATION Version 4.2 | 19 February 2018 |

Article 7 of Regulation (EU) No 1227/2011 (REMIT) stipulates that the Agency for the Cooperation of Energy Regulators ('the Agency') shall collect the data for assessing and monitoring wholesale energy markets as provided for in Article 8 of REMIT. The Agency shall ensure operational reliability of the information received pursuant to Article 8 of REMIT and it shall take all necessary measures to prevent any misuse of, and unauthorised access to, the information maintained in its systems as provided for in Article 12(1) of REMIT.

Pursuant to Article 8(1) of REMIT, market participants, or third parties on their behalf, shall provide the Agency with a record of wholesale energy market transactions. The European Commission shall, by means of implementing acts, adopt uniform rules on the reporting of records of transactions, including orders to trade ('trade data') pursuant to Article 8(2) of REMIT.

According to the implementing acts adopted by the Commission on 17 December 2014 as Commission Implementing Regulation (EU) No 1348/2014 and its Article 11(1), the Agency shall develop technical and organisational requirements for submitting data in order to ensure efficient, effective and safe exchange and handling of information.

For this purpose, the Agency developed and published Requirements for the registration of Registered Reporting Mechanisms (RRMs) - (RRM Requirements) which were consulted with relevant parties in 2013 and 2014 and adopted by the Agency on 7 January 2015. In the context of the drafting of the RRM requirements, the Agency consulted with relevant parties about the data validation rules for REMIT transaction reporting which were implemented in the Agency's REMIT Information System (ARIS). These rules do not represent any change to the REMIT transaction reporting rules or guidance, but merely tighten up the validation to ensure that the Agency does not receive erroneous transaction data from the industry for its market monitoring activities, and propagate that data to National Regulatory Authorities through the Agency's data sharing solution.

Whilst the data validation rules were initially only made available to RRMs as the actual target group of the data validation rules, the Agency has decided to make this document available to the general public for transparency reasons in order to facilitate the communication between RRMs and REMIT market participants on the data validation rules implemented in ARIS. This document should be read in combination with the ARIS Data Validation Rules Configuration document which is made available in parallel. Both documents will be updated as required. Data submissions should be aligned to the validations in these documents.

Please note that the target group of these documents remains unchanged. They continue to be addressed to RRMs. This is why the Agency will only accept questions on this document and/or the ARIS Data Validation Rules Configuration document submitted by RRMs. Market participants should send any questions they may have on the REMIT Information System Data Validation document and/or on the ARIS Data Validation Rules Configuration document to the RRMs reporting data on their behalf.

Table of Contents

| | |
|--|-----------|
| TABLE OF CONTENTS | 4 |
| 1 INTRODUCTION | 8 |
| 1.1 PURPOSE..... | 8 |
| 1.2 SCOPE..... | 8 |
| 1.3 TARGET AUDIENCE..... | 8 |
| 1.4 GLOSSARY..... | 8 |
| 2 DATA VALIDATION OVERVIEW | 9 |
| 2.1 DATA TYPES VALIDATED..... | 9 |
| 2.1.1 <i>Standard Contract Data Types</i> | 9 |
| 2.1.2 <i>Non-Standard Contract Data Types</i> | 9 |
| 2.1.3 <i>Fundamental Data Types</i> | 10 |
| 2.1.4 <i>Inside Information Data Types</i> | 10 |
| 3 HANDLING VALIDATION FAILURE | 11 |
| 3.1 REJECTION OF AN INVALID DATA SET..... | 11 |
| 3.2 REJECTION OF AN INVALID RECORD..... | 11 |
| 3.3 FLAGGING OF A RECORD WITH WARNING MESSAGE..... | 11 |
| 4 TECHNICAL DATA VALIDATION | 12 |
| 4.1 FILE NAMING CONVENTION VALIDATION..... | 12 |
| 4.1.1 <i>File Name Component Check</i> | 12 |
| 4.1.2 <i>Date Format Check</i> | 12 |
| 4.1.3 <i>Schema Name Check</i> | 13 |
| 4.1.4 <i>Valid Schema Version Check</i> | 13 |
| 4.1.5 <i>Party Identifier Check</i> | 13 |
| 4.1.6 <i>Sequence Number Check</i> | 13 |
| 4.2 SIGNATURE AND ENCRYPTION VALIDATION..... | 14 |
| 4.2.1 <i>Valid Encryption Type</i> | 14 |
| 4.2.2 <i>Data can be decrypted</i> | 14 |
| 4.2.3 <i>Signature Matches Participant</i> | 14 |
| 4.3 SCHEMA COMPLIANCE VALIDATION..... | 14 |
| 4.4 DATABASE SCHEMA COMPLIANCE..... | 15 |
| 4.5 DATA DUPLICATION CHECKS IN A SINGLE FILE..... | 15 |
| 4.5.1 <i>Standard / Supply Contract Duplicates Check</i> | 15 |
| 4.5.2 <i>Transportation Contract Duplicates Check</i> | 16 |
| 4.5.3 <i>Fundamental Data Duplicates Check</i> | 16 |
| 5 STANDARD CONTRACT FUNCTIONAL DATA VALIDATION | 18 |
| 5.1 FIELD INTEGRITY COMPLIANCE..... | 18 |
| 5.1.1 <i>Transaction Identification</i> | 18 |
| 5.1.2 <i>Party Identifier Validation</i> | 18 |
| 5.1.3 <i>Embedded Contract Integrity</i> | 18 |
| 5.2 FUNCTIONAL DATA QUALITY COMPLIANCE..... | 18 |
| 5.2.1 <i>Valid Price Validation</i> | 19 |
| 5.3 SEQUENCING AND LOGICAL DATA COMPLIANCE..... | 19 |
| 5.3.1 <i>Organised Market Place (EXC) Sequencing Data Compliance</i> | 19 |

| | | |
|----------|--|-----------|
| 5.3.2 | Organised Market Place (EXC) Logical Data Compliance | 20 |
| 5.3.3 | Organised Market Place (BRK) Sequence Data Compliance | 20 |
| 5.3.4 | Organised Market Place (BRK) Logical Data Compliance | 21 |
| 5.3.5 | Auction Sequence Data Compliance | 21 |
| 5.3.6 | Auction Logical Data Compliance | 22 |
| 5.3.7 | Non-Venue / Off Market or Bilateral Sequence Data Compliance | 22 |
| 5.3.8 | Non-Venue / Off Market or Bilateral Logical Data Compliance | 22 |
| 5.4 | DOUBLE REPORTING VALIDATION CHECKS | 22 |
| 5.5 | HOW TO CORRECT TRANSACTIONS | 23 |
| 5.6 | ANNEX TABLE 1 FIELD LEVEL VALIDATION RULES | 24 |
| 5.6.1 | Field No 21: Contract Id | 24 |
| 5.6.2 | Field No 44: Option Style | 25 |
| 5.6.3 | Field No 45: Option Type | 25 |
| 5.6.4 | Field No 47: Option Strike Price | 26 |
| 5.6.5 | Field No 48: Delivery Point or Zone | 26 |
| 5.7 | ANNEX TABLE 1 CROSS FIELD LEVEL VALIDATION RULES | 26 |
| 5.7.1 | Field No 1: ID of the market participant or counterparty | 26 |
| 5.7.2 | Field No 11: Buy/Sell Indicator | 27 |
| 5.7.3 | Field No 13: Order Id | 27 |
| 5.7.4 | Field No 21: Contract ID | 28 |
| 5.7.5 | Field No 25: Contract Name | 28 |
| 5.7.6 | Field No 29: Last Trading Date Time | 29 |
| 5.7.7 | Field No 35: Price | 30 |
| 5.7.8 | Field No 37: Price Currency | 31 |
| 5.7.9 | Field No 38: Notional amount | 31 |
| 5.7.10 | Field No 39: Notional Currency | 31 |
| 5.7.11 | Field No 40: Quantity / Volume | 32 |
| 5.7.12 | Field No 41: Total Notional Contract Quantity | 33 |
| 5.7.13 | Field No 49: Delivery Start Date | 33 |
| 5.7.14 | Field No 54: Load Delivery Intervals | 34 |
| 5.7.15 | Field No 57: Price/time interval quantity | 34 |
| 5.7.16 | Field No 58: Action type | 35 |
| 5.8 | ANNEX TABLE 1 EXTERNAL FIELD VALIDATION RULES | 40 |
| 5.8.1 | Field No 1: ID of the market participant or counterparty | 40 |
| 5.8.2 | Field No 6: Reporting entity ID | 40 |
| 5.8.3 | Field No 8: Beneficiary Identification | 41 |
| 5.8.4 | Field No 27: Organised market place ID/OTC | 41 |
| 5.9 | ANNEX TABLE 1 LOGICAL VALIDATION RULES | 42 |
| 5.9.1 | Field No 31: Additional UTI Info | 42 |
| 5.9.2 | Field No 58: Action Type | 42 |
| 5.9.3 | Parallel reporting channel | 43 |
| 5.9.4 | Back loading of historical data for Phase 1 reporting | 44 |
| 5.9.5 | Record Sequence Number check | 45 |
| 5.9.6 | Exceptions not linked to a specific rule: | 45 |
| 6 | NON-STANDARD CONTRACT FUNCTIONAL DATA VALIDATION | 47 |
| 6.1 | FIELD INTEGRITY COMPLIANCE | 47 |
| 6.1.1 | Delivery Product Validation | 47 |
| 6.1.2 | Participant Code Validation | 47 |
| 6.1.3 | Contract Identification Validation | 47 |

| | | |
|----------|--|-----------|
| 6.1.4 | <i>Delivery Profile Validation</i> | 47 |
| 6.2 | FUNCTIONAL DATA QUALITY COMPLIANCE | 47 |
| 6.2.1 | <i>Counterparty Identification</i> | 48 |
| 6.2.2 | <i>Matching Transaction Identifier</i> | 48 |
| 6.3 | ANNEX TABLE 2 FIELD LEVEL VALIDATION RULES | 48 |
| 6.3.1 | <i>Field No 1: ID of the market participant or counterparty</i> | 48 |
| 6.3.2 | <i>Field No 3: ID of the other market participant or counterparty</i> | 48 |
| 6.3.3 | <i>Field No 5: Reporting entity ID</i> | 49 |
| 6.3.4 | <i>Field No 15: Price or price formula</i> | 49 |
| 6.3.5 | <i>Field No 28 / No 29: First fixing date / Last fixing date</i> | 49 |
| 6.3.6 | <i>Field No 34 / No 35: Option first exercise date / Option last exercise date</i> | 50 |
| 6.3.7 | <i>Field No 41: Delivery point or zone</i> | 50 |
| 6.3.8 | <i>Field No 42 / No 43: Delivery start date / Delivery end date</i> | 50 |
| 6.3.9 | <i>Field No 45: Action Type</i> | 51 |
| 6.3.10 | <i>How to correct transactions for non standard contracts</i> | 52 |
| 6.3.11 | <i>Parallel reporting channel</i> | 52 |
| 6.3.12 | <i>Back loading of historical data for Phase 2 reporting</i> | 53 |
| 7 | TRANSPORTATION CONTRACT FUNCTIONAL DATA VALIDATION | 54 |
| 7.1 | FIELD INTEGRITY COMPLIANCE | 54 |
| 7.1.1 | <i>Delivery Product Validation</i> | 54 |
| 7.1.2 | <i>Participant Code Validation</i> | 54 |
| 7.1.3 | <i>Contract Identification Validation</i> | 54 |
| 7.1.4 | <i>Delivery Profile Validation</i> | 54 |
| 7.2 | FUNCTIONAL DATA QUALITY COMPLIANCE | 54 |
| 7.2.1 | <i>Counterparty Identification</i> | 55 |
| 7.2.2 | <i>Matching Transaction Identifier</i> | 55 |
| 7.3 | ANNEX TABLE 3 FIELD LEVEL VALIDATION RULES | 55 |
| 7.3.1 | <i>Field No 1 / No 2: Document identification / Document version</i> | 55 |
| 7.3.2 | <i>Field No 4 Sender Identification</i> | 55 |
| 7.3.3 | <i>Field No 6 Receiver Identification</i> | 56 |
| 7.3.4 | <i>Field No 10 Domain</i> | 56 |
| 7.3.5 | <i>Field No 19 In Area</i> | 57 |
| 7.3.6 | <i>Field No 20 Out Area</i> | 57 |
| 7.3.7 | <i>Back loading of historical data for Phase 2 reporting</i> | 58 |
| 7.4 | ANNEX TABLE 4 FIELD LEVEL VALIDATION RULES | 58 |
| 7.4.1 | <i>Document identification / Document version</i> | 58 |
| 7.4.2 | <i>Recipient Identification</i> | 59 |
| 7.4.3 | <i>Field No 1 Issuer Identification</i> | 59 |
| 7.4.4 | <i>Field No 2 Organised Market Place Identification</i> | 60 |
| 7.4.5 | <i>Field No 22 Connection point Identification</i> | 60 |
| 7.4.6 | <i>Field No 25 Responsible TSO Identification</i> | 61 |
| 7.4.7 | <i>Field No 26 Adjacent TSO Identification</i> | 61 |
| 7.4.8 | <i>Field No 27 Primary Market Participant Identification</i> | 61 |
| 7.4.9 | <i>Back loading of historical data for Phase 2 reporting</i> | 62 |
| 8 | FUNDAMENTAL DATA FUNCTIONAL DATA VALIDATION | 63 |
| 8.1 | FIELD INTEGRITY COMPLIANCE | 63 |
| 8.1.1 | <i>Participant Code Validation</i> | 63 |
| 8.1.2 | <i>Time Period Validation</i> | 63 |

| | | |
|-----------|--|-----------|
| 8.2 | FUNCTIONAL DATA QUALITY COMPLIANCE | 63 |
| 8.2.1 | <i>Geographical Location Representation Validation</i> | 63 |
| 8.2.2 | <i>Fundamental Data Type Validation</i> | 63 |
| 8.3 | FUNDAMENTAL FIELD DATA VALIDATION | 64 |
| 8.3.1 | <i>Valid EIC Code Format</i> | 64 |
| 8.3.2 | <i>Valid EIC Codes Received</i> | 64 |
| 8.3.3 | <i>No Duplicated Time Series</i> | 64 |
| 8.4 | FUNDAMENTAL FIELD LEVEL VALIDATION RULES | 64 |
| 8.4.1 | <i>Field: Reporting entity ID (sender_MarketParticipant.mRID / issuer_MarketParticipant.identification / reportingEntityIdentifier)</i> | 64 |
| 9 | DATA CORRECTION VALIDATION | 66 |
| 9.1 | CORRECTING AN EXISTING TRANSACTION VALIDATION | 66 |
| 10 | LATE REPORT TRANSACTION VALIDATION | 67 |
| 10.1 | DUPLICATE REPORTING TRANSACTION VALIDATION | 67 |
| 10.2 | TRANSACTION LINKING VALIDATION | 67 |

Annex I: Error conditions for Data Exchange Process

1 Introduction

1.1 Purpose

The purpose of this document is to identify the data validation rules which will be used by ACER to ensure the data quality provided by RRM.

1.2 Scope

The document covers the following points:

- Technical data quality validation
- Functional data quality validation
- Data corrections validation
- Validation of late reported transaction data

1.3 Target Audience

This document is targeted at ACER to ensure the approach for data validation of data collected by ARIS Tier 1 and Tier 2.

1.4 Glossary

RRM – Registered Reporting Mechanism

ENTSO – European Network of Transmission System Operators

XML – Extensible Markup Language

XSD – Extensible Schema Definition

CEREMP – Centralized European Register for Market Participants

EIC – Energy Identification Code

BIC – Banking Identification Code

LEI – Legal Entity Identifier

GS1/GLN – Global Location Number

DCI – Data Collection Instance

W3C – World Wide Web Consortium

ISO – International Organization for Standardization

2 Data Validation Overview

Data validation for data received from RRM is important to ensure that when analysed by ACER in Tier 3, and for use of shared data in Tier 4, it is of high enough quality to provide useful market monitoring and to allow ACER to perform surveillance.

There are two levels of data validation, an initial data validation performed on a technical level as part of the initial data quality checks from Tier 1 and a further in-depth analysis performed at the level of Tier 2 where integrity checks across data and reporting is performed.

2.1 Data Types Validated

ACER will be collecting different data types from RRM. The following section describes the data types that will be validated.

2.1.1 Standard Contract Data Types

Transactions relating to standard contracts for the supply / transportation and to derivative contracts will be reported by RRM, the majority of which will be transactional reports from organised market places (including auction exchanges and continuous market exchanges) or execution / broker venues.

A contract is a pre-defined or listed contract that is visible to all market participants and can be traded by any participant within the market place (subject the normal trading restrictions of the contract).

A contract is traded by participants submitting orders which result in trades or by accepting the terms of an order presented via screen trading which results in a trade. These orders can either be via an electronic exchange based trading or by some other medium of communication, written or verbal.

All data submitted for transactions on contracts will be submitted against a standardized contract transaction data report schema, which has a specific set of definitions for the technical submission of the standard contract transaction reports. When performing technical validation (see Technical Data Validation) data will be validated against this schema.

2.1.2 Non-Standard Contract Data Types

A non-standard contract will be reported by RRM, either on behalf of a single party or all parties of a contract. By definition, the non-standard contracts do not adhere to the normal contract definitions and represent trading on non-standard representations of products for delivery. These contracts will be specialised agreements between two or more participants that are negotiated outside of the normal visible market place and do not involve any other parties for the hosting or negotiation of the trade.

All data submitted for non-standard contracts will be submitted against a non-standard contract data schema, which has a specific set of definitions for the technical submission of the non-standard contract reports. Transactions which occur on non-standard contracts, i.e. the trades between two parties as a result of the non-standard contract will be reported using the standardized transaction report schema. When performing technical validation (see Technical Data Validation) data will be validated against this schema.

2.1.3 Fundamental Data Types

Fundamental data covers a wide range of categories of data, including all data that is submitted by RRM. It is likely that provision of fundamental data will be via the ENTSO organisations, either ENTSO-E (Electricity) or ENTSO-G (Gas), however, some data will be reported by individual TSOs. These organisations provide a consolidated transparency platform for all data relating to the transmission of electricity or gas, including data relating to the production, consumption, transportation of electricity or gas and the storage of natural gas and LNG and transportation of LNG.

In addition to the data provided by ENTSO organisations, fundamental data also covers data which impacts the overall physical market characteristics, such as weather data.

All data submitted by RRM will be based on the type of fundamental data provided. Each data type has a schema defined for the data content that will be provided to support the validation of technical submission of this data. When performing technical validation (see Technical Data Validation), data will be validated against these schemas.

2.1.4 Inside Information Data Types

Insider information data covers any urgent market news or announcements that may impact the utilisation of or the price of energy products, such as identification of outages or changes in consumption patterns of wholesale participants.

All insider information will be published by market participants via a variety of data sources. ACER will need to collect this data from these sources. There is likely to be no direct submission by RRM for this type of data, so there is limited conformance or validation specified for this data type.

3 Handling Validation Failure

The failure of a validation can result in several different outcomes:

3.1 Rejection of an Invalid Data Set

If an invalid data set is provided, i.e. the data is technically non-compliant with the specified data format and therefore can't be interpreted by the system, the data set is archived and a notification of rejection for the entire data set is sent to the RRM.

3.2 Rejection of an Invalid Record

If an invalid record is rejected, it will be archived and a notification of rejection for that particular record is sent to the RRM.

Invalid records should be shared with NRAs if they can be allocated to the respective NRAs according to the rules laid down in the data sharing agreement.

3.3 Flagging of a Record with warning message

A record that does not overcome specific validation rules can be flagged in the database with a warning message. The flag contains a reference to the relevant validation rule so that the flag of a record is comprehensible. The record is not considered invalid and no error notification is sent to the RRM.

4 Technical Data Validation

Technical data validation applies to all data types. Each report can be technically validated against the schema defined for the data type and technically against the data contained within the supplied report.

4.1 File Naming Convention Validation

All files submitted by market participants must be initially validated for the file naming convention as defined in the technical specifications. The following validations shall be performed against the file naming convention.

4.1.1 File Name Component Check

Does the file have the correct number of components in accordance with the file naming convention? The file naming convention has separation by the character ‘_’ and contains fields for each of the following components;

- a) Date of Submission of data
- b) Schema Name
- c) Schema Version
- d) Party Identifier
- e) Sequence Number

This means that each file should have the correct number of identifiers within the file name to correctly identify the file.

4.1.2 Date Format Check

The date specified should represent the date (in UTC) of the submission of the data set by the reporting participant.

The date format check validates that the file has been supplied with a file name that has the first component which matches the date schema YYYYMMDD and that the parts YYYY equate to a valid year, MM equates to a valid month and DD equates to a valid day of the month MM.

For example, using 20130815, the validation would check that 20130815 matches the schema YYYYMMDD, that 08 is a valid entry for MM and that 15 is a valid entry for DD and that 15 is a valid number of days in MM.

If 20130230 was received, this would be rejected as MM 02 only has a maximum of 28 (or 29 in a leap year) number of days, so this is not a valid date and would therefore fail the validation.

4.1.3 Schema Name Check

Each of the schemas are pre-defined for submission, if the schema name doesn't match with one of the pre-defined schema naming codes, then the naming convention check will fail.

For example, the naming code for the schema for standard contracts is REMITTable1_V1, if there is a file received with
YYYYMMDD_REMITTable1_V1_ACERCode_SeqNum.xml.asc.pgp
(e.g. 20150123_REMITTable1_V1_A0000037O.AT_1.xml.asc.pgp), then the REMITTable1_V1 would pass the validation as a valid schema name. Any name that doesn't match would be rejected and would not be handled.

4.1.4 Valid Schema Version Check

Each schema that is handled by the DCI has a valid version for definition; this is an integer number, i.e. 1 to 999. The check will fail based on two principles;

- a) The version number is not a valid version number, i.e. 1A or 0.1 is not a valid version number
- b) The schema specified isn't a supported version, for example, if a schema only has up to version 2, but version 3 is specified, then this would fail the validation.

4.1.5 Party Identifier Check

All of the RRM's will be registered with the ARIS system. During the registration process the RRM will obtain a recognised code with ACER, i.e. the ACER code. If the party identifier check doesn't reconcile the party code in the naming convention supplied, the file will be rejected from entry to the database and the party will be notified that corrective action must be taken, i.e. correct submission using their registered identifier.

4.1.6 Sequence Number Check

Each file supplied must be provided in the correct sequence, i.e. numerically increasing by 1 on each increment. If a file is not received in the correct sequence within a single batch upload, it will be an invalid upload and the RRM will be asked to submit the missing files. Note that the sequence numbers are unique per date of submission and per RRM. The check on the sequence depends on the data exchange interface:

Web Gui or Web Services: if the RRM uploads a file in the wrong sequence (for example n. 1,2 and then 5) the interface will display an error message explaining that the sequence number is wrong; in the above example the RRM must upload files n. 3, 4 and then can actually submit number 5.

Using SFTP: if the RRM uploads a file in the wrong sequence (for example n.1, 2 and then 5), the system will generate a receipt error for file n.5 that will not be usable anymore for all

interfaces. The submission sequence will now require submission of file n.3, subsequently n.4 and then n. 6 (sequence n.5 must be skipped since that sequence id has been marked as an error).

If for some reason a submitted file (for example n.3 in the sequence) is rejected due to technical failure (e.g. XML corrupted or not compliant with the schema), then the RRM must correct the original XML file, rename the file using the next relevant sequence number for the day of submission.

4.2 Signature and Encryption Validation

All data submitted by a RRM must be encrypted in alignment with the encryption key provided by ACER as part of the public-private key pairing. In addition to the data encryption, ACER will require that the data submitted is also electronically signed by the RRM's private key.

ACER will perform three tests at Tier 1 on the data provided by the RRM's:

4.2.1 Valid Encryption Type

Each file must be identifiable as a valid encryption type.

4.2.2 Data can be decrypted

The file must be decrypted upon receipt of the file; if it fails to decrypt using the specified encryption type, the file will be rejected.

4.2.3 Signature Matches Participant

Each RRM has a unique signature key, shared with ACER. If the party identifier specified in the file doesn't match the signature associated with the delivered file, then the file is rejected.

In this case the whole file will be rejected and the participant will have to resubmit all records and ensure that the file complies with the technical file publishing criteria.

4.3 Schema Compliance Validation

Each schema definition sets out the rules that must be adhered to in relation to the compliance of the schema.

An initial pass of all XML files delivered by the participant will be performed to validate the compliance with the schema definition.

The W3C schema definition (<http://www.w3.org/2001/XMLSchema>) is used as a basis for validating that all XML files delivered are compliant with the schemas.

This means that any typed fields or patterns within the schema will need to be adhered to by the participant, with any failures in the compliance validation causing the file to be rejected.

For example, if a date field is specified and all date fields must be ISO compliant in UTC, then any field which is a date field must comply with the standard.

In the case of technical failure to comply with the schema, the whole file will be flagged as invalid and the RRM will have to resubmit the rejected file.

4.4 Database Schema Compliance

The schema compliance validation only provides a technical definition of compliance for the actual schema; however, it doesn't ensure integrity between dependent fields within the schema. The technical data compliance will ensure that any interdependencies within the data of a single record are compliant with the technical specification.

For example, the field market participant identifier is defined by a code, but it is also associated with a code type. This means that if a participant defines a code as the market participant and defines that the code is represented as an LEI code, then the participant code must be a compliant LEI code.

So if the participant specifies LEI code for the market participant, then if the market participant code doesn't comply with the equivalent ISO 17442 standard and contain 20 unique alphanumeric characters, then the values are rejected as invalid and the data is not compliant.

Or another example; if a field is designated as a numerical field, such as the quantity of a contract, if the field doesn't contain a number, then the field will be rejected.

For fundamental data, the rules for technical compliance also apply, however the field definitions are provided by the ENTSO's (ENTSO-E and ENTSO-G) and they are responsible for ensuring that the correct definitions are included within the schemas.

In case of records with validation errors, those records will be flagged as invalid and the RRM will have to resubmit the corrected records in a new file.

4.5 Data Duplication Checks in a Single File

All records provided by a participant must be unique based on a unique key and the data provided within the record. If a duplicate record is provided within a single file entry by the RRM, the last duplicated record will be flagged as invalid; then the RRM will have to check the reason for duplication and take relevant actions according to the data submitted.

4.5.1 Standard / Supply Contract Duplicates Check

A duplicate record for supply contracts (REMIT Table 1) will be identified by the following fields being duplicated within the published data set:

- Contract Identifier and Organised Market Place Identifier
- Order / Trade Identifier (including Linked Order Identifier)
- Order / Trade Side Identifier
- Transaction Lifecycle (Action Type)
- Transaction Type (Order/Trade)

4.5.2 Transportation Contract Duplicates Check

The duplication of data provided by the ENTSO-E or ENTSO-G schemas is defined and managed by the ENTSOs as they are responsible for the definition of the data schemas and the data delivered through those schemas.

A duplicate record for transportation of electricity contracts will be identified by the following fields being duplicated within the published data set.

- Identification
- Revision Number
- Type
- Period / Time Interval
- Sender Market Participant

This applies to any of the document elements that can be reported using the schemas defined for Annex Table 3 registered with the IEC defined as iec62325-451-3-rights, iec62325-451-3-bid and iec62325-451-3-totalallocation.

A duplicate record for transportation of gas contracts will be identified by the following fields being duplicated within the published data set

- Identification
- Version
- Type
- Validity Period
- Issuer Market Participant Identification

This applies to any of the document elements that can be reported using the data schemas for Annex Table 4 created by Edig@s defined as urn-easee-gas-eu-edigas-remit-gascapacityallocationsdocument for the purpose of reporting for REMIT.

4.5.3 Fundamental Data Duplicates Check

The duplication of data provided by the ENTSO-E or ENTSO-G schemas is defined and managed by the ENTSOs as they are responsible for the definition of the data schemas and the data delivered through those schemas.

A duplicate record for transportation of electricity contracts will be identified by the following fields being duplicated within the published data set.

- Identification
- Revision Number / Version
- Type
- Validity Period / Time Interval
- Sender / Issuer Market Participant

5 Standard Contract Functional Data Validation

The following detailed validation checks will be applied to transactions on standard contracts only, i.e. any contract which is traded through an organised market place, an exchange venue or through registered channels on a contract which is recognised and registered with ACER.

5.1 Field Integrity Compliance

Functional validation of the field integrity ensures that all fields that are linked at a functional level are validated. There are multiple test cases in this that will need to be performed; (each test case can be enhanced or additional checks added when issues are found with existing data):

5.1.1 Transaction Identification

The ID specified for lifecycle changes (e.g. modification record) matches an ID for a trade execution as performed on the organised market place. If no match exists then the record is flagged as invalid.

5.1.2 Party Identifier Validation

In this validation, any field that identifies a registered participant with ACER, for example the market participant field will be validated against the real values within the database to ensure that the values reported actually exist. If a party field is incorrectly reported, then the transaction needs to be flagged to ensure that the party is correctly identified.

In this case the record will be identified as having a specific field failure and the participant will have to resubmit the record as a data correction. Failures of verifications will cause the data to be flagged in the database with an error code and highlighted as a failure to the RRM.

5.1.3 Embedded Contract Integrity

Orders and trades should always have a reference to a specific contract; the contract can be embedded in the order / trade or can be included in the contract list. If an order or trade is invalid because a validation rule is not passed, then the existing embedded contract will be invalidated as well.

5.2 Functional Data Quality Compliance

Functional data quality identifies a more in depth data quality check to ensure consistency between fields and within the data itself.

5.2.1 Valid Price Validation

Each transaction should have a valid price defined, i.e. the monetary value put on the volume unit of the transaction. If the price field doesn't have a valid value, or the value is outside of the deemed normal or regular values, validation will fail.

For example, it is not sensible / or doesn't appear to be sensible to have a price of contract that exceeds the maximum or minimum defined values, i.e. 99999999.99999 or -99999999.99999.

The price is defined in the schema as:

- Total digits: 20
- Fraction digits: 5

Failures of these verifications will cause the rejection of the submitted file.

5.3 Sequencing and Logical Data Compliance

Each transaction must be sequenced correctly according to the platform / engine the events occurred in and comply with the logical representation of the transactions in relation to how the events occurred in real time. For example, when a participant submits an order to an organised market place, the order can result in a trade or the order can be amended or deleted, but none of the events can occur prior to the order being originally submitted to the organised market place.

If a participant reports transactions which break these rules then the transactions will be identified as failures and must be corrected.

The following sections define the compliance rules that must be / will be validated to ensure that the data quality reported is sufficient for ACER to be able to perform market monitoring and market surveillance.

An Organised Market Place, which under Article 3 of the implementing acts has a market place type of EXC is defined as an exchange where parties submit orders which are matched by the market place for trading.

An Organised Market Place under Article 3 of the implementing acts with a market place type of BRK is defined as a broker platform of execution where parties may submit orders for trade and use electronic or voice traded facilities to agree on trades.

5.3.1 Organised Market Place (EXC) Sequencing Data Compliance

1. All events must be time sequenced based on the time of occurrence in the organised market place.

2. All orders must be sequenced with a lifecycle event 'new' as the first event in the lifecycle of the order.
3. No order can produce a trade event unless the order has previously been identified in the sequence, i.e. the order must exist for it to trade.
4. No order can be modified if it has not already been identified using a 'new' event in the lifecycle of the order.
5. No order can be cancelled if it has not already been identified using a 'new' event in the lifecycle of the order.
6. No order can be modified or cancelled if the order has fully traded by a trade or multiple trade events, i.e. if the order has no volume to trade and has been fully filled, it can't trade.

Note: In the event that the system cancels the trade and resubmits the order automatically, i.e. in the case of a cancelled auction result and a new auction, the order resubmission should be represented as a 'new' event for the purpose of reporting with the new time of submission and new gate closure date and time.

5.3.2 Organised Market Place (EXC) Logical Data Compliance

1. Order events can only trade with order events from the opposite side of the market place, i.e. a buy order can only trade with a sell order.
2. Buy and Sell orders must match according to the price-time priority of the market place, i.e. if a buy order doesn't match the price of the sell order it won't trade, this must be reflected in the transactions.
3. Order and trade volumes must allow a match; for example, it is incorrect to represent a trade which has more volume on it than is visible or available on the order.

5.3.3 Organised Market Place (BRK) Sequence Data Compliance

1. All events must be time sequenced based on the time of occurrence in the venue traded.
2. All orders must be sequenced with a lifecycle event 'new' as the first event in the lifecycle of the order, this includes order events that are agreed by some other medium of communication, written or verbal.

3. No order can produce a trade event unless the order has previously been identified in the sequence, i.e. the order must exist for it to trade. Where a trade is agreed via verbal or external communication and it is traded on screen, an order record should be created to record this. This is already the case for broker trading platforms; however, this behaviour is unknown for other exchange venues and may be reconsidered if it significantly impacts the venue. If the order is traded off screen (i.e. purely by voice), then the on screen order must be reported as cancelled and the trade reported as a voice traded transaction.
4. No order can be modified if it has not already been identified using a 'new' event in the lifecycle of the order.
5. No order can be cancelled if it has not already been identified using a 'new' event in the lifecycle of the order.
6. No order can be modified or cancelled if the order has fully traded by a trade or multiple trade events, i.e. if the order has no volume to trade, it can't trade.

5.3.4 Organised Market Place (BRK) Logical Data Compliance

1. Click trade events can only trade with order events from the opposite side of the market place, i.e. a buyer can only trade with a sell order, the order part that has not been entered to the system should be recorded in the trade report.
2. Buy and Sell orders match according to the preference set out by participants and agreements; there is no verification of priority of trading.
3. Order and trade volumes must match. For example, it is incorrect to represent a trade which has more volume on it than is visible or available on the order. If a change in a volume of an order, prior to the trade, has been agreed externally to the venue, this should be recorded as a step in the lifecycle of the order and a transaction reported for this.

5.3.5 Auction Sequence Data Compliance

1. All events within the auction should be time sequenced in accordance with when the events occurred.
2. All orders submitted to an implied auction must be represented before the resultant trading of the auction.
3. Trades as a result of an implied auction must all be represented at the time at which the auction occurred. The trades should be represented in the priority order assigned by the auction.

5.3.6 Auction Logical Data Compliance

1. All orders submitted within a single auction bid schedule (i.e. their price levels and volumes are intrinsically linked) must be represented as a single order report containing each price and volume point.
2. All trades as a result of the auction are not directly associated with a submitted bid schedule by a participant but are as a result of the auction, trades prices and volumes may not be directly linked to the order submitted by the same participant, but are a resultant of the auction matching.

5.3.7 Non-Venue / Off Market or Bilateral Sequence Data Compliance

1. Trade agreements that do not have orders placed by participants should be represented as trade events which are time sequenced.
2. A market participant must report their side of the trade, with the matching identifier for the other participant included in the data.

5.3.8 Non-Venue / Off Market or Bilateral Logical Data Compliance

1. All trades should identify the accurate price and volume of the trade
2. All trades should identify the link between the sides of the trades and the participants involved in the trading
3. Trade sides must be matched with the counterparty trade side; a trade side will not be published into Tier 2 until it has been confirmed by the counterparty. The counterparty must provide a matching identifier for the transaction.

In case the record does not comply with the above described rules it will be identified as having a specific field failure and the RRM will have to resubmit the record as a data correction.

5.4 Double Reporting Validation Checks

This validation will occur against transactions that have been submitted by RRMs that have reported transactions for organised market places or trading venues which are registered as reporting participants themselves. For example, RRM(1) may submit transactions on behalf of all trading participants on their system, however if a participant reports their transactions with RRM (2) as well, this will cause double reporting.

A unique way of identifying the transactions for participants shall be used to identify the duplicate records.

A duplicate record will be identified by the following fields being duplicated within the record:

- Contract Identifier
- Order / Trade Identifier (including Linked Order Identifier)
- Order / Trade Buy/Sell Indicator
- Action Type
- Organised Market Place /Identifier
- **HasSubstring**(field Extra, 'FullSet')¹

If all of these fields are duplicated, then the transaction is deemed to be a duplicate transaction.

In this case it will be checked if the suspected duplicate record is exactly the same as the record which is in the database.

The system will flag as a duplicate the last processed transaction and only one RRM (who reported the last processed transaction) will be notified that the transaction was reported as a duplicate. The duplicate transaction will be imported into the database and flagged as invalid.

5.5 How to correct transactions

When an XML file contains a trade with ActionType = 'E', all previously loaded trades (in other XML files) with the same UTI / ContractID / Organised Market Place / IdOfMarketParticipant / LinkedOrderID / **HasSubstring**(field Extra, 'FullSet') and with transactionTime = transactionTime of trade with actionType = 'E' are logically deleted.

Important notice: *if the received Trade with ActionType = 'E' does not match any existing trade with the same UTI, ContractID, Organised Market Place, idOfMarketParticipant, LinkedOrderID, **HasSubstring**(field Extra, 'FullSet') and transactionTime then the received trade is invalid.*

- *Error code: R1LIATTRNOETRAN*
- *Error message: No Trade found*

In case of trade transactions with errors, the RRM must send an XML file containing the erroneous trades with ActionType = 'E'. If the RRM wants to correct previously sent trades it should include in the same XML file the "corrected" trades.

¹ The notation used here describes a Boolean function that returns *true* if the field Extra includes the substring 'FullSet' (case insensitive) and *false* otherwise. If two records are received which differ only in the value of this function it is assumed that the record for which the value is *false* has been reported on behalf of the Market Participant. The field Extra should be populated with values 'FullSet==Yes;omp==yes' for REMITTable1_V2 or 'FullSet_Yes' for REMITTable1_V1. It is expected that this field is only filled by OMPs reporting also the data (e.g. as part of a full set of data covering all trading activity for a particular day) that may not necessarily be reported on behalf of the Market Participants.

When an XML file contains an order with `ActionType = 'E'`, all previously loaded orders (in other XML files) with the same “OrderId / ContractID / Organised Market Place / IdOfMarketParticipant” / **HasSubstring**(field Extra, 'FullSet') and with `transactionTime = transactionTime` of order with `actionType = 'E'` are logically deleted.

Important notice: *If the received Order with `ActionType = 'E'` does not match any existing order with the same Order ID, Contract ID, Organised Market Place, idOfMarketParticipant, **HasSubstring**(field Extra, 'FullSet') and `transactionTime`, then the received order is invalid.*

- *Error code: R1LIATORNOETRAN*
- *Error message: No Order found*

In case of order transactions with errors, the RRM must send an XML file containing the erroneous orders with `ActionType = 'E'`. If the RRM wants to correct previously sent orders it should include in the same XML file the “corrected” orders.

5.6 Annex Table 1 Field Level Validation Rules

Transactions which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

5.6.1 Field No 21: Contract Id

The rules for this field do not apply to transaction reports provided for bilateral trading, identified by the value XBIL in the Organised Market Place Field 27.

Rule 1: Every order report or trade report received must identify the contract that the order report or trade report occurs against, either referring to the contract id as defined in the contract list or by providing the contract identification as part of the order or trade report.

Each uploaded XML file is identified by a load ID. If orders or trades (not containing full contract definitions) do not have a reference to a specific contract available in the Contract List, then the order or trade is invalid.

Rule 2ACTCIR1

*If the pair (Order.load_ID, Order.contractID) does not match a pair (Contract.loadID, Contract.ContractID) and **OMP != XBIL** then the Order is invalid*

- *Error code: R1CTCITRAIDE*
- *Error message: Transaction does not reference to an existing Contract*

*If the pair (Trade.load_ID, Trade.contractID) does not match a pair (Contract.loadID, Contract.ContractID) and **OMP != XBIL** then Trade is invalid*

- *Error code: R1CTCITRAIDE*

- *Error message: Transaction does not reference to an existing Contract*

5.6.2 Field No 44: Option Style

Rule 1: A contract with a contract type defined as OP, OP_FU, OP_FW or OP_SW must also have a value defined in the field “option style”.

Rule 2AODOSR1

If Contract type= ‘OP|OP_FU|OP_FW|OP_SW’ AND Option Style is NOT defined then the Contract is INVALID

- *Error code: R1ODOSOPTM*
- *Error message: Contract Type ‘OP|OP_FU|OP_FW|OP_SW’ must have Option Style defined*

5.6.3 Field No 45: Option Type

Rule 1: A contract with a contract type defined as OP, OP_FU, OP_FW or OP_SW must also have a value defined in the field “option type”

Rule 2AODOTR1

If Contract type= ‘OP|OP_FU|OP_FW|OP_SW’ AND Option Type is NOT defined then the Contract is INVALID

- *Error code: R1ODOTOPTM*
- *Error message: Contract Type OP, OP_FU, OP_FW or OP_SW must have Option Type defined*

Rule 2: All values in the value option exercise date must be prior to the delivery start date of the delivery profile

Rule 2AODOEDR2

If Contract.DeliveryStartDate <= Contract.optionExerciseDate” then the Contract is INVALID.

- *Error code: R2ODOEDOPT*
- *Error message: Contract Option Exercise Date must be prior to the Delivery Start Date*

Rule 2AODOEDR2x

If Contract.DeliveryStartDate <= Contract.optionExerciseDate” and OptionStyle is != OTH (‘O’) then the Contract is INVALID.

- *Error code: R2ODOEDOPT*

- *Error message: Contract Option Exercise Date must be prior to the Delivery Start Date*

5.6.4 Field No 47: Option Strike Price

Rule 2AODOSPR1

Rule 1: A contract with a contract type defined as OP, OP_FU, OP_FW or OP_SW must also have a value defined in the field “option strike price”

If Contract type=’OP|OP_FU|OP_FW|OP_SW’ AND Option Strike Price is NOT defined then the Contract is INVALID

- *Error code: R1ODOSPOPT*
- *Error message: Contract Type OP, OP_FU, OP_FW or OP_SW must have Option Strike Price defined*

5.6.5 Field No 48: Delivery Point or Zone

Rule 1: The delivery zone must be defined as EIC codes identifying delivery points or zones. For delivery points or zones where multiple delivery points are possible, then more than one delivery point must be provided

Rule 2ADPDPOZR1

All “deliveryPointOrZone” EIC must be present in “EIC Registry” available in ARIS.

If Contract.deliveryPointOrZone is NOT in the List of Accepted EICs (TRUM ANNEX VI) then the Contract is INVALID.

- *Error code: R1DPDPZMUL*
- *Error message: Delivery Point or Zone is not registered*

5.7 Annex Table 1 Cross Field Level Validation Rules

Transactions which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

5.7.1 Field No 1: ID of the market participant or counterparty

Rule 1: The trade report must identify the market participant for the order report linked to the trade report, the market participant must be represented as an ACER code for the analysis, which means the system will normalise the identifier to ACER code from the supplied value by the submitted transaction.

Rule 2BPCIMPCPR1

If Trade.LinkedReaderId = Order.OrderId and Trade.ContractID = Order.ContractID and Order.transactionTime <= Trade.transactionTime then if Trade.idOfMarketParticipant != Order.idOfMarketParticipant for the given reportingEntityID (the comparison key must include the reporting entity identifier), then the Trade is INVALID. The system will invalidate also the related linked Orders inside the same LOAD_ID.

- *Error code: R1PTCMPCOIT*
- *Error message: Market Participant for trade does not match Market Participant for order*

5.7.2 Field No 11: Buy/Sell Indicator

Rule 1: All trades produced by an organised market place must have a buy and sell side identified by unique characteristics.

Rule 2BPCBSIR1

If Trade.buySellIndicator != 'B' AND Trade.buySellIndicator != 'S' then the Trade is INVALID.

- *Error code: R1PTCBSIOMPUQ*
- *Error message: Trade with invalid buy/sell Indicator*

Rule 2: All continuous market orders (i.e. orders placed for intention to trade, not for implied auction) shall have an indication of buy or sell intention. All order reports that have a contract type indicating the contract is traded continuously, shall have either a buy or sell indicator value

Rule 2BPCBSIR2

If Order.buySellIndicator = 'C' AND Contract.ContractType != 'AU' then Order is INVALID

- *Error code: R2PTCBSICMOM*
- *Error message: Order with invalid buy/sell Indicator*

5.7.3 Field No 13: Order Id

Rule 1: All order reports have a defined contract identifier which matches the defined contract list or contains full contract details

Rule 2ACNTRR1

If the pair (Order.load_ID, Order.contractID) does not match a pair (Contract.loadID, Contract.ContractID) then the Order is INVALID

- *Error code: R1CNTRTRAIDE*
- *Error message: Transaction does not reference an existing Contract*

If the pair (Trade.load_ID, Trade.contractID) does not match a pair (Contract.loadID, Contract.ContractID) then Trade is INVALID

- *Error code: R1CNTRTRAIDE*
- *Error message: Transaction does not reference an existing Contract*

5.7.4 Field No 21: Contract ID

Rule 1: The organised market place must have only a single contract identifier for the contract for the lifetime of the tradable contract. The contract identifier must be unique and is not shared with any other contract

Rule 2: The contract identifier provided for the identification of the contract must be unique for the contract and must only identify the single tradable contract. This applies across venues that trade the same listed contract

Rule 3: The contract must always be represented using the same contract identifier throughout the life of the contract, i.e. up to the date of delivery

Rule 4: If the Organised Market Place ID is “XBIL” then the Contract ID field must contain “NA”

Rule 2ACTCIR2

If Contract(loadID, ContractID, OrganisedMarketPlaceID, CL=True) is NOT unique then those contracts are INVALID.

N.B.: If CL(Contract List) = True then the Contract is inside the Contract List, if CL = false the Contract is embedded (inline).

- *Error code: R2CTCIOMPUNQ*
- *Error message: Duplicated contract in Contract List*

Rule 2BCCONIDXR1

If the Organised Market Place ID is “XBIL” AND If Contract ID != “NA” then the record is INVALID.

- *Error code: 2BCCONIDXE1*
- *Error message: Invalid contract ID for a bilateral contract*

5.7.5 Field No 25: Contract Name

Rule 1: If the Organised Market Place ID is “XBIL” and the Contract ID field contains “NA” then the Contract name field must have a match against “BILCONTRACT”, “BACKLOADING” or “EXECUTION” values.

Rule 2BCCONNMXR1

If the Organised Market Place ID is “XBIL” AND Contract ID = “NA” AND Contract name is NOT “BILCONTRACT”, “BACKLOADING” or “EXECUTION” then the record is INVALID.

- Error code: 2BCCONNMXE1
- Error message: Invalid contract name for bilateral contract

5.7.6 Field No 29: Last Trading Date Time

Rule 1: If the contract last trading date is specified after the start of the delivery date, then the report is invalid.

Rule **2ACTCIR6**

If (Not (Contract.EnergyCommodity = NG and (DeliveryEndDate – DeliveryStartDate = 1 day))) and Contract.LastTradingDateTime is NOT empty AND Contract.LastTradingDateTime > Contract.DeliveryStartDate then Contract is INVALID;

- Error code: R6CLTDTCDST
- Error message: Contract last trading time greater than contract delivery start date

Rule **2ACTCIR3**

If Contract.LastTradingDateTime is NOT empty AND Order.TransactionTime > Contract.LastTradingDateTime then Order is INVALID;

- Error code: R2CLTDTOT
- Error message: Transaction timestamp greater than last trading time

If Contract.LastTradingDateTime is NOT empty AND Contract.Type != ‘AU’ AND Trade.TransactionTime > Contract.LastTradingDateTime then Trade is INVALID;

- Error code: R2CLTDTOT
- Error message: Transaction timestamp greater than last trading time

Rule **2ACTCIR4**

If (Not (Contract.EnergyCommodity = NG and (DeliveryEndDate – DeliveryStartDate = 1 day))) and Contract.LastTradingDateTime is empty AND Order.ActionType=New AND Order.TransactionTime > Contract.DeliveryStartDate then Order is INVALID;

- Error code: R2CLTDTDSTOT
- Error message: Transaction timestamp greater than contract delivery start date

If (Not (Contract.EnergyCommodity = NG and (DeliveryEndDate – DeliveryStartDate = 1 day))) and Contract.LastTradingDateTime is empty AND Trade.ActionType=New AND Trade.TransactionTime > Contract.DeliveryStartDate then Trade is INVALID;

- Error code: R2CLTDTDSTOT
- Error message: Transaction timestamp greater than contract delivery start date

Rule **2BTDTRADE**

The following check will be applied ONLY when TerminationDate will be available in TRADE

If Trade.TerminationDate is NOT empty AND Trade.TerminationDate > CONTRACT.DeliveryEndDate then Trade is Invalid

- Error code: R2TRTDCONDED
- Error message: Trade termination date greater than contract delivery end date

5.7.7 Field No 35: Price

Rule 2: All continuous market orders (i.e. orders placed for intention to trade, not for implied auction) shall have a single price. All order reports that have a contract type indicating the contract is traded continuously, shall have a defined price

Rule **2BCDPR2**

(If OrderReport ->price is defined and OrderReport -> priceIntervalQuantityDetails -> priceTimeIntervalQuantity -> value (Field No 57) is defined)

OR

(If Order.orderType != 'MAR' and Contract.contract_type != AU and OrderReport ->price is NOT defined and OrderReport -> priceIntervalQuantityDetails -> priceTimeIntervalQuantity -> value (Field No 57) is NOT defined)

Then the Order is INVALID

- Error code: R2CDPRCMOSP
- Error message: Order price undefined or Order price defined both at OrderReport level and at priceIntervalQuantityDetails level

Rule **2BCDPTR2**

(If TradeReport ->price is defined and TradeReport -> priceIntervalQuantityDetails -> priceTimeIntervalQuantity -> value (Field No 57) is defined)

OR

(If TradeReport ->price is NOT defined and TradeReport -> priceIntervalQuantityDetails -> priceTimeIntervalQuantity -> value (Field No 57) is NOT defined)

Then the Trade is INVALID

- *Error code: R2CDPRCMTSP*
- *Error message: Trade price undefined or Trade price defined both at TradeReport level and at priceIntervalQuantityDetails level*

5.7.8 Field No 37: Price Currency

Rule 1: All trades should match based on the same price currency so buy side trade report and sell side trade report must have the same price currency value

Rule 2BCDPCR1

If a received Trade has buySellIndicator = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with buySellIndicator = 'S' (or 'B') and both have (actionType = 'N') AND (Sell side TRADE.price_currency) != (Buy side TRADE.price_currency) then a warning message linked to Trade is saved in database.

- *Warning code: R1CDPCBSCM*
- *Warning message: Trade with invalid price currency*

5.7.9 Field No 38: Notional amount

Rule 1: All trades should match at the same notional amount so buy side trade report and sell side trade report must have the same notional amount. Exception is auction trades where volumes for trading is aggregated and may not equally match individual buy and sell transactions

Rule 2BCDNAR1

If a received Trade has buySellIndicator = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with buySellIndicator = 'S' (or 'B') and both have (actionType = 'N') AND (ContractType != 'AU') AND (Sell side TRADE.notional_amount) != (Buy side TRADE.notional_amount) then a warning message linked to Trade is saved in database.

- *Warning code: R1CDNANABSAM*
- *Warning message: Trade with invalid notional amount*

5.7.10 Field No 39: Notional Currency

Rule 1: All trades should match based on the same notional currency so buy side trade report and sell side trade report must have the same notional currency value

Rule 2BCDNCR1

If the received Trade has buySellIndicator = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with buySellIndicator = 'S' (or 'B') and both have

(actionType = 'N') AND (Sell side TRADE.notional_currency) != (Buy side TRADE.notional_currency) then a warning message linked to Trade is saved in database.

- Warning code: R1CDNCBSTSC
- Warning message: Trade with invalid notional currency

5.7.11 Field No 40: Quantity / Volume

Rule 1: All trades should match at the same volume, so buy side trade report and sell side trade report must have the same volume. Exception is auction trades where volumes for trading are aggregated and may not equally match individual buy and sell orders.

Rule 2BCDQVR1

If a received Trade has buySellIndicator = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with buySellIndicator = 'S' (or 'B') and both have (actionType = 'N') AND (contractType != 'AU') AND (Sell side trade.Quantity) != (Buy side trade.Quantity) then a warning message linked to Trade is saved in database.

- Warning code: R1CDQVBSTSV
- Warning message: Trade with invalid quantity

Rule 2: All trades must have a volume which is not zero, i.e. a volume for a trade must be bought or sold

Rule 2BCDQVR2

(If TradeReport.quantity is defined and TradeReport -> priceIntervalQuantityDetails -> quantity (Field No 55) is defined)

OR

((If TradeReport.quantity is NOT defined OR TradeReport.quantity = 0) AND (TradeReport -> priceIntervalQuantityDetails -> quantity is NOT defined OR TradeReport -> priceIntervalQuantityDetails -> quantity = 0) AND ContractName is NOT "EXECUTION")

then Trade is INVALID

- Error code: R2CDQVNZ
- Error message: Trade with invalid quantity or Trade quantity defined both at TradeReport level and at priceIntervalQuantityDetails level

Rule 3: All continuous market orders (i.e. orders placed for intention to trade, not for implied auction) shall have a single volume. All order reports that have a contract type indicating the contract is traded continuously, shall have a defined volume

Rule 2BCDQVR3

(If OrderReport.quantity is defined and OrderReport -> priceIntervalQuantityDetails -> quantity (Field No 55) is defined)

OR

(If OrderReport.quantity is NOT defined AND OrderReport -> priceIntervalQuantityDetails -> quantity is NOT defined AND Order.contractType != AU)

then the Order is INVALID

- *Error code: R3CDQVCMSV*
- *Error message: Order quantity undefined OR Order quantity defined both at OrderReport level and at priceIntervalQuantityDetails level.*

5.7.12 Field No 41: Total Notional Contract Quantity

Rule 1: All trades should match at the same notional quantity, so buy side trade report and sell side trade report must have the same notional quantity. Exception is auction trades where volumes for trading is aggregated and may not equally match individual buy and sell transactions

Rule 2BCDTNCQR1

If received Trade has buySellIndicator = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with buySellIndicator = 'S' (or 'B') and both have (actionType = 'N') AND (ContractType != 'AU') AND (Sell side Trade.totalNotionalContractQuantity) != (Buy side Trade.totalNotionalContractQuantity) *then a warning message linked to Trade is saved in database.*

- *Warning code: R1CDTNCQBSSM*
- *Warning message: Trade with invalid Total Notional Quantity*

5.7.13 Field No 49: Delivery Start Date

Rule 1: The delivery profile must specify a start and end date; the end date of delivery must be after the start date. If the delivery is only for a single day, then the delivery end date should be equal to the delivery start date

Rule 2ADPDSDR1

If Contract.DeliveryStartDate > Contract.DeliveryEndDate and the Contract is INVALID

- *Error code: R1DPDEDCHK*
- *Error message: Contract start date greater than contract end date*

5.7.14 Field No 54: Load Delivery Intervals

Rule 1: The delivery interval start time and end time must define the end time after the start time. For more than one interval within a day, the intervals should be defined in chronological sequence and no intervals should overlap.

Rule 2ADPLDIR1

All intervals must be chronological sequenced:

If `Contract.loadDeliveryStartTime` > `Contract.loadDeliveryEndTime` the Contract is INVALID

N.B.: if `Contract.loadDeliveryStartTime` = `Contract.loadDeliveryEndTime` = 00:00 then the Contract is VALID because the EndTime is considered 24:00

- Error code: R1DPLDINTCHK
- Error message: Load delivery start time greater **than load delivery end time**

Rule 2ADPLDIR1

All intervals must not overlap:

If `Contract.loadDeliveryEndTime` (previous interval) > `Contract.loadDeliveryStartTime` (current interval) the contract is INVALID

- Error code: R2DPLDINTCHK
- Error message: Load delivery end time overlaps next load delivery start time

5.7.15 Field No 57: Price/time interval quantity

Rule 1: The price defined by the order must allow matching of the trade to occur. I.e. for the defined interval, the specified price for buying must be equal or more than the price for selling.

Rule 2BDPPTIQR1

If the received Trade has `buySellIndicator` = 'B' (or 'S') and there is a matching Trade (based on UTI and ContractID) on the other side with `buySellIndicator` = 'S' (or 'B') and both have (`actionType` = 'N') AND (`Sell side Trade.priceTimeIntervalQuantity`) != (`Buy side Trade.priceTimeIntervalQuantity`) then a warning message linked to Trade is saved in database.

- Warning code: R1DPPTIQORTRM
- Warning message: Trade Price Time Interval Quantity invalid

5.7.16 Field No 58: Action type

Rule 1: All transaction reports are originally created with action type “new”, i.e. the first instance of an order/trade transaction chain must always have a value “new”.

Rule **2BLIATR1**

If the last received Trade has `actionType = 'N'` and there is an existing Trade with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant`) then the last received Trade is INVALID.

- *Error code: R1LIATTRNEW*
- *Error message: Received a duplicate Trade Report in Submission*

Rule **2BLIATR1x**

*If the last received Trade has `actionType = 'N'` and there is an existing Trade with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant`, `LinkedOrderID` and **HasSubstring**(field Extra, 'FullSet')) then the last received Trade is INVALID.*

- *Error code: R1LIATTRNEW*
- *Error message: Received a duplicate Trade Report in Submission*

Rule **2BLIATR1**

If the last received Order has `actionType = 'N'` and there is an existing Order with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `OrderID`, `idOfMarketParticipant`) then the last received Order is INVALID.

- *Error code: R1LIATORNEW*
- *Error message: Received a duplicate Order Report in Submission*

Rule **2BLIATR1x**

*If the last received Order has `actionType = 'N'` and there is an existing Order with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `OrderID`, `idOfMarketParticipant` and **HasSubstring**(field Extra, 'FullSet')) then the last received Order is INVALID.*

- *Error code: R1LIATORNEW*
- *Error message: Received a duplicate Order Report in Submission*

Rule **2DLIATR2**

If the last received Order has `actionType = 'M'` and there is no existing Order with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant`) then the last received Order is **INVALID**.

- Error code: `R1LIATORMOD`
- Error message: Received an Order Modification for an Order that doesn't exist in the system

Rule **2DLIATR2x**

If the last received Order has `actionType = 'M'` and there is no existing Order with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant` and **HasSubstring**(field `Extra`, '`FullSet`')) then the last received Order is **INVALID**.

- Error code: `R1LIATORMOD`
- Error message: Received an Order Modification for an Order that doesn't exist in the system

Rule **2DLIATR3**

If the last received Trade has `actionType = 'M'` and there is no existing Trade with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant`) then the last received Trade is **INVALID**.

- Error code: `R1LIATTRMOD`
- Error message: Received a Trade Modification for a Trade that doesn't exist in the system

Exception to the rule: if `OrganisedMarketPlaceID = XBIL` (bilateral trades) and `previousUniqueTransactionIdentifier` (or `additionalUTIInfo` in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule **2DLIATR3x**

If the last received Trade has `actionType = 'M'` and there is no existing Trade with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `UTI`, `idOfMarketParticipant`, `LinkedOrderID` and **HasSubstring**(field `Extra`, '`FullSet`')) then the last received Trade is **INVALID**.

- Error code: `R1LIATTRMOD`
- Error message: Received a Trade Modification for a Trade that doesn't exist in the system

Exception to the rule: if OrganisedMarketPlaceID = XBIL (bilateral trades) and previousUniqueTransactionIdentifier (or additionalUTIInfo in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule 2BLIATR2

If the last received Trade has actionType = 'C' and there is no existing Trade with actionType='N' and the same values for (buySellIndicator, ContractID, OrganisedMarketPlaceID, UTI, idOfMarketParticipant) then the last received Trade is INVALID.

- *Error code: R1LIATTRCAN*
- *Error message: Received a Trade Cancelled for a Trade that doesn't exist in the system*

Exception to the rule: if OrganisedMarketPlaceID = XBIL (bilateral trades) and previousUniqueTransactionIdentifier (or additionalUTIInfo in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule 2BLIATR2x

*If the last received Trade has actionType = 'C' and there is no existing Trade with actionType='N' and the same values for (buySellIndicator, ContractID, OrganisedMarketPlaceID, UTI, idOfMarketParticipant, LinkedOrderID and **HasSubstring**(field Extra, 'FullSet')) then the last received Trade is INVALID.*

- *Error code: R1LIATTRCAN*
- *Error message: Received a Trade Cancelled for a Trade that doesn't exist in the system*

Exception to the rule: if OrganisedMarketPlaceID = XBIL (bilateral trades) and previousUniqueTransactionIdentifier (or additionalUTIInfo in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule 2BLIATR2

If the last received Order has actionType = 'C' and there is no existing Order with actionType='N' and the same values for (buySellIndicator, ContractID, OrganisedMarketPlaceID, OrderID (or PreviousOrderID), idOfMarketParticipant) then the last received Order is INVALID.

- *Error code: R1LIATORCAN*
- *Error message: Received an Order Cancelled for an Order that doesn't exist in the system*

Rule 2BLIATR2x

If the last received Order has *actionType* = 'C' and there is no existing Order with *actionType*='N' and the same values for (*buySellIndicator*, *ContractID*, *OrganisedMarketPlaceID*, *OrderID* (or *PreviousOrderID*), *idOfMarketParticipant* and **HasSubstring**(field Extra, 'FullSet')) then the last received Order is INVALID.

- Error code: R1LIATORCAN
- Error message: Received an Order Cancelled for an Order that doesn't exist in the system

Rule **2DLIATR5**

If the last received Trade has *actionType* = 'E' and there is no existing Trade with *actionType*='N' and the same values for (*buySellIndicator*, *ContractID*, *OrganisedMarketPlaceID*, *UTI*, *idOfMarketParticipant*) then the last received Trade is INVALID.

- Error code: R1LIATTRERR
- Error message: Received a Trade Error for a Trade that doesn't exist in the system

Exception to the rule: if *OrganisedMarketPlaceID* = XBIL (bilateral trades) and *previousUniqueTransactionIdentifier* (or *additionalUTIInfo* in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule **2DLIATR5x**

If the last received Trade has *actionType* = 'E' and there is no existing Trade with *actionType*='N' and the same values for (*buySellIndicator*, *ContractID*, *OrganisedMarketPlaceID*, *UTI*, *idOfMarketParticipant*, *LinkedOrderID* and **HasSubstring**(field Extra, 'FullSet')) then the last received Trade is INVALID.

- Error code: R1LIATTRERR
- Error message: Received a Trade Error for a Trade that doesn't exist in the system

Exception to the rule: if *OrganisedMarketPlaceID* = XBIL (bilateral trades) and *previousUniqueTransactionIdentifier* (or *additionalUTIInfo* in REMIT Table1 V2) is NOT empty then the above rule is not applicable.

Rule **2DLIATR5**

If the last received Order has *actionType* = 'E' and there is no existing Order with *actionType*='N' and the same values for (*buySellIndicator*, *ContractID*, *OrganisedMarketPlaceID*, *OrderID* (or *PreviousOrderID*), *idOfMarketParticipant*) then the last received Order is INVALID.

- Error code: R1LIATORERR
- Error message: Received an Order Error for an Order that doesn't exist in the system

Rule **2DLIATR5x**

If the last received Order has `actionType = 'E'` and there is no existing Order with `actionType='N'` and the same values for (`buySellIndicator`, `ContractID`, `OrganisedMarketPlaceID`, `OrderID` (or `PreviousOrderID`), `idOfMarketParticipant` and **HasSubstring**(field Extra, 'FullSet')) then the last received Order is INVALID.

- Error code: R1LIATORERR
- Error message: Received an Order Error for an Order that doesn't exist in the system

Rule **2BODOIR3**

If the received Order has `actionType = C` and there is an already existing Order with same (`OrderID`, `ContractID`, `buySellIndicator` and `OrganisedMarketPlaceID`) and `actionType = C` and the existing Order is NOT logically deleted (`actionType='E'`), then the last received Order is invalid.

- Error code: R2ODOICIMPIUQC
- Error message: Order with action type C duplicated

Rule **2BODOIR3x**

If the received Order has `actionType = C` and there is an already existing Order with same (`OrderID`, `ContractID`, `buySellIndicator` and `OrganisedMarketPlaceID` and **HasSubstring**(field Extra, 'FullSet')) and `actionType = C` and the existing Order is NOT logically deleted (`actionType='E'`), then the last received Order is invalid.

- Error code: R2ODOICIMPIUQC
- Error message: Order with action type C duplicated

Rule **2ACTUTIR2**

If the received Trade has the same (`UTI`, `ContractID`, `OrganisedMarketPlaceID`, `IdOfMarketParticipant`, `actionType = 'C'`, `BuySellIndicator`) of an already existing Trade then the received Trade record is invalid.

N.B. If the existing Trade is marked as logically deleted because of a previous Trade with `actionType = "E"`, then the above check is not applied.

- Error code: R1CDUTIDRCIMPDTUQC
- Error message: Duplicated trade: a trade with same UTI, ContractID, Organised Market Place Identifier, IdOfMarketParticipant and `actionType = 'C'` already exists

Rule 2ACTUTIR2x

If the received Trade has the same (UTI, ContractID, OrganisedMarketPlaceID, IdOfMarketParticipant, actionType = 'C', BuySellIndicator, LinkedOrderID and HasSubstring(field Extra, 'FullSet')) of an already existing Trade then the received Trade record is invalid.

N.B. If the existing Trade is marked as logically deleted because of a previous Trade with actionType = "E", then the above check is not applied.

- *Error code: R1CDUTIDRCIMPDTUQC*
- *Error message: Duplicated trade: a trade with same UTI, ContractID, Organised Market Place Identifier, IdOfMarketParticipant and actionType = 'C' already exists*

5.8 Annex Table 1 External Field Validation Rules

Transactions which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

5.8.1 Field No 1: ID of the market participant or counterparty

Rule 1: The Market Participant (MP) ID must have a match in CEREMP. A MP must be registered with NRAs (and available in CEREMP) and this field has to be validated against a real value within the CEREMP to ensure that the value reported actually exists.

Rule 2CPCIDOMPR1

If Market Participant (MP) ID of order/trade is NOT available in CEREMP DB than order/trade is INVALID.

- *Error code: R1REOMPCON*
- *Error message: Invalid Id of Market Participant*

Notice: this rule is not applicable in the Test Framework environment.

5.8.2 Field No 6: Reporting entity ID

Rule 1: A reporting entity ID must be supplied by all reporters.

ReportingEntityID is checked in ARIS application during validation of uploaded file, value for ReportingEntityID = RRM.

Rule 2: A reporting entity has to be registered with ACER before submitting data and the code has to match the value defined in ARIS

Rule 2CREREDR1

The Reporting Entity ID available in the XML file must be present in ARIS database; otherwise all contracts and transactions are invalid

- *Error code: R1REPENTCON*
- *Error message: Invalid Id of Reporting Entity*

Notice: this rule is not applicable in the Test Framework environment.

5.8.3 Field No 8: Beneficiary Identification

Rule 1: If the beneficiary is supplied, the beneficiary ID should have a valid ACER code and should have a match in the CEREMP system.

Rule 2CPCBIR1

If beneficiary ID is not in CEREMP then the order or trade beneficiary is identified for registration and listed in ARIS as not registered.

- *Warning code: R1BENMPCON*
- *Warning message: Invalid Id of Beneficiary*

Notice: this rule is not applicable in the Test Framework environment.

5.8.4 Field No 27: Organised market place ID/OTC

The OMP field is validated against the real values within the database to ensure that the values reported actually exist.

Rule 1: The organised market place (OMP) ID must have a match against the ACERs List of OMPs (published on the REMIT portal). An OMP must be registered with ACER and this field has to be validated against a real value within the ARIS to ensure that the value reported actually exists.

Rule 2CIDOOMPR1

If Organised market place (OMP) ID of order/trade is NOT available in ARIS DB than order/trade is INVALID.

- *Error code: R1INVOMPID*
- *Error message: Invalid Id of Organised market place*

Notice: this rule is not applicable in the Test Framework environment.

5.9 Annex Table 1 Logical Validation Rules

Transactions which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

5.9.1 Field No 31: Additional UTI Info

Notice: this rule applies only to the **REMIT Table1 version 2 schema**.

Preamble: to overcome the problem of the UTI for coupled intraday market that share the same order book we have amended the name of the field “previousUniqueTransactionIdentifier” to “additionalUtilInfo” which will capture the system ID hosting the shared order book making the UTI unique.

Rule 2BADUTIINF

If OrganisedMarketPlaceId <> XBIL and additionalUtilInfo is NOT empty the uniqueness of a trade report is based on the following condition: *If the last received Trade has actionType = ‘N’ and there is an existing Trade with actionType=‘N’ and the same values for (buySellIndicator, **additionalUTInfo**, OrganisedMarketPlaceID, UTI, idOfMarketParticipant) then the last received Trade is INVALID.*

- Error code: R1ADUTIINF
- Error message: Received a duplicate Trade Report in Submission

5.9.2 Field No 58: Action Type

Rule 1: An order must not be deleted by “cancel” action prior to the trade report

Rule 2DLIATR1

If the last received Trade has Trade.TransactionTime (with Action.Type N or M) > Order.TransactionTime of an already existing Order with the same ContractId, Organised Market Place, IdOfMarketParticipant with Order.actionType = ‘C’, for the given reportingEntityID (the comparison key must include the reporting entity identifier), then Trade is INVALID.

- Error code: R1LIATDECAPR
- Error message: Invalid Trade

Rule 3: A transaction can’t have a lifecycle action update indicating a “modified” state if the previous update is defined as “cancelled”

Rule 2DLIATR6

If the last received Order has ActionType ‘M’ and there is an existing Order with the same OrderId, ContractID, Organised Market Place, IdOfMarketParticipant and with actionType = ‘C’

AND last received Order.transactionTime greater than existing Order.transactionTime then the Order is invalid.

- Error code: R6LIAORNOMODAFCAN
- Error message: Invalid Order

Rule **2DLIATR6**

If the last received Trade has ActionType 'M' and there is an existing Trade with the same UTI, ContractID, Organised Market Place, IdOfMarketParticipant and same BuySellerIndicator with actionType = 'C' AND last received Trade.transactionTime greater than existing Trade.transactionTime then the Trade is invalid.

- Error code: R6LIATTRNOMODAFCAN
- Error message: Invalid Trade

Rule 4: A transaction can't have a lifecycle action update indicating a "new" state if the previous update is defined as "cancelled"

Rule **2DLIATR7**

If the last received Order has ActionType 'N' and there is an existing Order with the same OrderId, ContractID, Organised Market Place, IdOfMarketParticipant and with actionType = 'C' AND last received Order.transactionTime greater than existing Order.transactionTime then the Order is invalid.

- Error code: R7LIAORNONEWAFCAN
- Error message: Invalid Order

Rule **2DLIATR7**

If the last received Trade has ActionType 'N' and there is an existing Trade with the same UTI, ContractID, Organised Market Place, IdOfMarketParticipant and BuySellerIndicator with actionType = 'C' AND last received Trade.transactionTime greater than existing Trade.transactionTime then the Trade is invalid.

- Error code: R7LIATTRNONEWAFCAN
- Error message: Invalid Trade

5.9.3 Parallel reporting channel

To report an update to a previously submitted:

- a) back-loaded record due to a lifecycle event ('M' – modification, 'C' – cancellation, 'E' – error) or
- b) previously accepted live record due to a lifecycle event ('M' – modification, 'C' – cancellation, 'E' – error),

that currently would not pass the configured validation rules due to the original record's content violating the currently enabled validation rules, a parallel reporting channel for submission is available.

Records submitted using the parallel reporting channel must be included in a file beginning with "20000101_". All active validation rules will be applied to all records that are present in the submitted file.

If the date in the filename of the submitted XML file is equal to "20000101" and if there are one or more records (orders and/or trades) in the submitted XML file with `actionType = 'N'` then the input file will be rejected:

- *Rejected Technical*
- *Error Code: 94*
- *Error Description: Invalid Date Failure*

From DCI web GUI (User uploads menu) the following message will be displayed:

- *Unallowed Actiontype N for parallel reporting channel*

5.9.4 Back loading of historical data for Phase 1 reporting

Rule 1: the system will allow back loading of historical data skipping any validation rule.

Rule *BLRDTTTR1*

If the date in the filename of the submitted XML file is prior (<) to the Date1 (5 Oct 2015 00:00:00Z), then all validation rules (except the ones defined in this section) will be ignored.

If any trade record in the submitted XML file with the date in the filename prior (<) to the Date1 (5 Oct 2015 00:00:00Z) has a `TransactionTime >=` the Date2 (7 Oct 2015 00:00:00Z), then the trade record is rejected;

- *Error code: E1BLRDTTT*
- *Error message: Expected back loading data received at least one live data*

if there are one or more order records in the submitted XML file, then these order records are rejected.

- *Error code: E1BLRD TTO*
- *Error message: Orders are not accepted for back loading data*

Rule *BLRDTTTR2*

If the date in the filename of the submitted XML file is greater or equal to Date1 (5 Oct 2015 00:00:00Z) and there exists a trade record with a `TransactionTime <` Date1 (5 Oct 2015

00:00:00Z), then trade record will be rejected (notice: orders can have a transactionTime < Date1).

- *Error code: E2BLRDTT*
- *Error message: Expected live data received at least one back loading data*

Note: Backloading has been discontinued since 30 June 2017

5.9.5 Record Sequence Number check

According to the schema definition every order / trade report must have a unique record sequence number (RecordSeqNumber);

Rule 1: this rule will enforce the record sequence number starting from Min[RecordSeqNumber] and monotonically increasing without any gaps to Max[RecordSeqNumber].

Rule SCMSCRSNR1

The record sequence numbers must match the following formula for trades and orders separately:

$$(\text{Min}[\text{RecordSeqNumber}] + \text{Max}[\text{RecordSeqNumber}]) * (\text{Max}[\text{RecordSeqNumber}] - \text{Min}[\text{RecordSeqNumber}] + 1) / 2 = \text{Sum}[\text{RecordSeqNumber}]$$

If the order/trade records do not comply with the above rule the following error message will be generate for all order/trade records:

- *Error code: E1SCMSCRSN*
- *Error message: Record Sequence Number must be monotonic ascending without gaps*

5.9.6 Exceptions not linked to a specific rule:

If the received Order, or Trade, is invalid and it has an embedded contract then the embedded contract is invalidated.

- *Error code: R1CONEMBINVAL*
- *Error message: Embedded Contract is invalidated*

Important notice: NO error message is sent back to the RRM since the validation error is related to the order or trade.

If the received Order is invalid and it has an embedded leg contract then the embedded leg contract is invalidated.

- *Error code: R1LEGCONEMBINVAL*
- *Error message: Embedded Leg Contract was invalidated*

Important notice: NO error message is sent back to the RRM since the validation error is related to the order.

If the received Contract (in the Contract List, embedded, or embedded leg contract) is invalid, all Orders and Trades related to it are invalidated.

- *Error code: R1CONINVORD*
- *Error message: Order with invalid related Contract*
- *Error code: R1CONINVTRA*
- *Error message: Trade with invalid related Contract*

Important notice: an error message is sent back to the RRM for orders or trades related to invalid contracts.

6 Non-Standard Contract Functional Data Validation

Non-standard contracts have a more limited validation set than that of standard contracts.

6.1 Field Integrity Compliance

Functional validation of the field integrity ensures that all fields that are linked at a functional level are validated. There are multiple test cases in this that will need to be performed; (each test case can be enhanced or additional checks added when issues are found with existing data):

6.1.1 Delivery Product Validation

In this validation, the data provided by the market participant will be checked to ensure that the products to be delivered have been correctly identified and that there is adequate description of the product for ACER to determine the details of the contract.

6.1.2 Participant Code Validation

In this validation, any field that identifies a registered participant with ACER; for example the market participant field or the reporting participant field, will be validated against the real values within the database to ensure that the values reported actually exist. If a participant field is incorrectly reported, then the transaction needs to be highlighted to ensure that the participant is correctly identified.

6.1.3 Contract Identification Validation

Whilst non-standard contracts are not registered directly with ACER; ACER will keep track of any contracts that have been reported and product codes which are assigned to these by participants will be used to validate any existing contracts. Where a duplicate product identifier is provided, ACER will identify the duplicated product code. This may then be assessed for standardisation of the product.

6.1.4 Delivery Profile Validation

Participants creating non-standard contracts must define the delivery profile of the product, i.e. they must have a defined delivery period and a defined delivery location. A delivery location or delivery profile must adhere to known zones or regions within the EU, for example a product which is delivered in a zone in France must be identified as a product being delivered in France.

6.2 Functional Data Quality Compliance

Because non-standard contracts are reported by participants who are involved in the transaction, either themselves or through a third party, there are several important elements in

the reporting that need to be validated to ensure that non-standard contracts can be married together correctly.

6.2.1 Counterparty Identification

All non-standard contracts must clearly identify the counterparty information on the trade transaction reported. This means that the correct participant information must be populated with a value that is recognised by ACER and the participant must be registered with ACER

6.2.2 Matching Transaction Identifier

When agreeing a contract between the two counterparties, they must agree on the definition of the contract and a way to identify this contract between the two parties. This contract identifier must be identified to ACER as part of the transaction identification to allow ACER to marry the reporting contract details from each counterparty.

6.3 Annex Table 2 Field Level Validation Rules

Contract reports which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

6.3.1 Field No 1: ID of the market participant or counterparty

Rule 1: The ID of the Market Participant must have a match in CEREMP. A MP must be registered with NRAs (and available in CEREMP) and this field has to be validated against a real value within the CEREMP to ensure that the value reported actually exists.

Rule AT2F1R1

If NonStandardContractReport.IdOfMarketParticipant is NOT in CEREMP database then the contract report is invalid

- *Error code: AT2F1E1*
- *Error message: Id of Market Participant is unknown*

Notice: this rule is not applicable in the Test Framework environment.

6.3.2 Field No 3: ID of the other market participant or counterparty

Rule 1: The ID of the other Market Participant must have a match in CEREMP. A MP must be registered with NRAs (and available in CEREMP) and this field has to be validated against a real value within the CEREMP to ensure that the value reported actually exists.

Rule AT2F3R1

If NonStandardContractReport.otherMarketParticipant is NOT in CEREMP database then the contract report is invalid

- *Error code: AT2F3E1*
- *Error message: Id of other Market Participant(s) is(are) unknown*

Notice: this rule is not applicable in the Test Framework environment.

6.3.3 Field No 5: Reporting entity ID

Rule 1: The ID of the Reporting Entity must have a match in the list of RRM's; furthermore the ID of the Reporting Entity must correspond to the ACER code used in the name of the submitted file.

For example, if the submitted filename is 20150715_REMITTable2_V1_A0000116N.UK_1.xml.asc.pgp, then the ID of the Reporting entity must correspond to the ACER code **A0000116N.UK**.

Rule ReportingEntityNNS

If ReportingEntityID is NOT in the list of RRM's

OR

ReportingEntityID does not correspond to the ACER code used for submitting the file,

then the contract report is invalid

- *Error code: R1REPENTCON*
- *Error message: Invalid Id of Reporting Entity*

6.3.4 Field No 15: Price or price formula

Rule 1: At least one between price and price formula must be defined

Rule AT2F15R1

If NonStandardContractReport -> priceOrPriceFormula -> price is NOT defined AND NonStandardContractReport -> priceOrPriceFormula -> priceFormula is NOT defined, then the record is invalid.

- *Error code: AT2F15E1*
- *Error message: Both price and priceFormula are undefined*

6.3.5 Field No 28 / No 29: First fixing date / Last fixing date

Rule 1: Last fixing date must be equal or greater than First fixing date.

Rule AT2F28R1

If NonStandardContractReport -> fixingIndexDetails -> firstFixingDate greater than NonStandardContractReport -> fixingIndexDetails -> lastFixingDate, then the record is invalid.

- *Error code: AT2F28E1*
- *Error message: First fixing date greater than Last fixing date*

6.3.6 Field No 34 / No 35: Option first exercise date / Option last exercise date

Rule 1: Option last exercise date must be equal or greater than Option first exercise date.

Rule AT2F34R1

If NonStandardContractReport -> optionDetails -> optionFirstExerciseDate greater than NonStandardContractReport -> optionDetails -> optionLastExerciseDate, then the record is invalid.

- *Error code: AT2F34E1*
- *Error message: Option first exercise date greater than Option last exercise date*

6.3.7 Field No 41: Delivery point or zone

Rule 1: The delivery point or zone must be defined as EIC codes identifying delivery points or zones. For delivery points or zones where multiple delivery points or zones are possible, then more than one delivery point or zone must be provided.

Rule AT2F41R1

If NonStandardContractReport -> deliveryPointOrZone is NOT on the List of Accepted EICs (TRUM ANNEX VI) then the Contract is INVALID.

- *Error code: AT2F41E1*
- *Error message: Delivery Point or Zone is not registered*

6.3.8 Field No 42 / No 43: Delivery start date / Delivery end date

Rule 1: Delivery end date must be equal or greater than Delivery start date.

Rule AT2F42R1

If NonStandardContractReport -> deliveryStartDate greater than NonStandardContractReport -> deliveryEndDate, then the record is invalid.

- *Error code: AT2F42E1*
- *Error message: Delivery start date greater than Delivery end date*

6.3.9 Field No 45: Action Type

Rule 1: All contract reports are originally created with action type “new”, i.e. the first instance of a contract report chain must always have a value “new”.

Rule AT2F45R1

If the last received contract report has actionType = ‘N’ and there is an existing contract report with actionType=‘N’ and the same values for (buySellIndicator, ContractID, idOfMarketParticipant, otherMarketParticipant(s)) then the last received contract report is INVALID.

- Error code: AT2F45E1
- Error message: Duplicated Contract Report for Action Type “new”

Rule 2: A contract report can be terminated with action type “cancel”, i.e. the last instance of a contract report chain can have a value “cancel”.

Rule AT2F45R2

If the last received contract report has actionType = ‘C’ and there is an existing contract report with actionType=‘C’ and the same values for (buySellIndicator, ContractID, idOfMarketParticipant, otherMarketParticipant(s)) then the last received contract report is INVALID.

- Error code: AT2F45E2
- Error message: Duplicated Contract Report for Action Type “cancel”

Rule AT2F45R3

If the last received contract report has actionType = ‘M’ and there is no existing contract report with actionType=‘N’ and the same values for (buySellIndicator, ContractID, idOfMarketParticipant, otherMarketParticipant(s)) then the last received contract report is INVALID.

- Error code: AT2F45E3
- Error message: Received a Contract Modification but its status New is missing.

Rule AT2F45R4

If the last received contract report has actionType = ‘C’ and there is no existing contract report with actionType=‘N’ and the same values for (buySellIndicator, ContractID, idOfMarketParticipant, otherMarketParticipant(s)) then the last received contract report is INVALID.

- Error code: AT2F45E4
- Error message: Received a Contract Cancellation but its status New is missing.

Rule **AT2F45R5**

If the last received contract report has *actionType* = 'E' and there is no existing contract report with *actionType*='N' and the same values for (*buySellIndicator*, *ContractID*, *idOfMarketParticipant*, *otherMarketParticipant(s)*) then the last received contract report is **INVALID**.

- Error code: *AT2F45E5*
- Error message: *Received a Contract Error Correction but its status New is missing.*

6.3.10 How to correct transactions for non standard contracts

When an XML file contains a contract with *ActionType* = 'E', all previously loaded contracts (in other XML files) with the same (*buySellIndicator*, *ContractID*, *idOfMarketParticipant*, *otherMarketParticipant(s)*) and with *contractDate* = *contractDate* of contract with *actionType* = 'E' are logically deleted.

Important notice: if the received Contract with *ActionType* = 'E' does not match any existing Contract with the same (*buySellIndicator*, *ContractID*, *idOfMarketParticipant*, *otherMarketParticipant(s)*) and *contractDate* then the received Contract is invalid.

- Error code: *AT2F45E6*
- Error message: *No Contract found (to apply the E correction action)*

In case of non std. contract transactions with errors, the RRM must send an XML file containing the erroneous non std. contracts with *ActionType* = 'E'. If the RRM wants to correct previously sent non std. contracts it should include in the same XML file the “corrected” non std. contracts.

6.3.11 Parallel reporting channel

To report an update to a previously submitted:

- c) back-loaded record due to a lifecycle event ('M' – modification, 'C' – cancellation, 'E' – error) or
- d) previously accepted live record due to a lifecycle event ('M' – modification, 'C' – cancellation, 'E' – error),

that currently would not pass the configured validation rules due to the original record's content violating the currently enabled validation rules, a parallel reporting channel for submission is available.

Records submitted using the parallel reporting channel must be included in a file beginning with "20000101_". All active validation rules will be applied to all records that are present in the submitted file.

If the date in the filename of the submitted XML file is equal to “20000101” and if there are one or more records (orders and/or trades) in the submitted XML file with `actionType = 'N'` then the input file will be rejected:

- *Rejected Technical*
- *Error Code: 94*
- *Error Description: Invalid Date Failure*

From DCI web GUI (User uploads menu) the following message will be displayed:

- *Unallowed Actiontype N for parallel reporting channel*

6.3.12 Back loading of historical data for Phase 2 reporting

The following rules apply for Non-Standard Contracts.

Rule 1: the system will allow back loading of historical data skipping any validation rule.

Rule AT2F12BLR1

If the date in the filename of the submitted XML file is prior (<) to the Date1 (5 Oct 2015 00:00:00Z), then all validation rules (except the ones defined in this section) will be ignored.

If any record in the submitted XML file with the date in the filename prior (<) to the Date1 (5 Oct 2015 00:00:00Z) has a **contractDate** >= the Date3 (7 Apr 2016 00:00:00Z), then the trade record is rejected;

- *Error code: AT2F12BLE1*
- *Error message: Expected back loading data received at least one live data*

Rule AT2F12BLR2

If the date in the filename of the submitted XML file is greater or equal to Date1 (5 Oct 2015 00:00:00Z) and there exists a trade record with a **contractDate** < Date4 (4 Apr 2016 00:00:00Z), then trade record will be rejected.

- *Error code: AT2F12BLE2*
- *Error message: Expected live data received at least one back loading data*

Note: Backloading has been discontinued since 30 June 2017

7 Transportation Contract Functional Data Validation

Transportation contracts have a more limited validation set than that of standard contracts.

7.1 Field Integrity Compliance

Functional validation of the field integrity ensures that all fields that are linked at a functional level are validated. There are multiple test cases in this that will need to be performed; (each test case can be enhanced or additional checks added when issues are found with existing data):

7.1.1 Delivery Product Validation

In this validation, the data provided by the market participant will be checked to ensure that the products to be delivered have been correctly identified and that there is adequate description of the product for ACER to determine the details of the contract.

7.1.2 Participant Code Validation

In this validation, any field that identifies a registered participant with ACER; for example the market participant field or the reporting participant field, will be validated against the real values within the database to ensure that the values reported actually exist. If a participant field is incorrectly reported, then the transaction needs to be highlighted to ensure that the participant is correctly identified.

7.1.3 Contract Identification Validation

Whilst non-standard contracts are not registered directly with ACER; ACER will keep track of any contracts that have been reported and product codes which are assigned to these by participants will be used to validate any existing contracts. Where a duplicate product identifier is provided, ACER will identify the duplicated product code. This may then be assessed for standardisation of the product.

7.1.4 Delivery Profile Validation

Participants creating non-standard contracts must define the delivery profile of the product, i.e. they must have a defined delivery period and a defined delivery location. A delivery location or delivery profile must adhere to known zones or regions within the EU, for example a product which is delivered in a zone in France must be identified as a product being delivered in France.

7.2 Functional Data Quality Compliance

Because transportation contracts are reported by participants who are involved in the transaction, either themselves or through a third party, there are several important elements in

the reporting that need to be validated to ensure that non-standard contracts can be married together correctly.

7.2.1 Counterparty Identification

All non-standard contracts must clearly identify the counterparty information on the trade transaction reported. This means that the correct participant information must be populated with a value that is recognised by ACER and the participant must be registered with ACER

7.2.2 Matching Transaction Identifier

When agreeing a contract between the two counterparties, they must agree on the definition of the contract and a way to identify this contract between the two parties. This contract identifier must be identified to ACER as part of the transaction identification to allow ACER to match the reporting contract details from each counterparty.

7.3 Annex Table 3 Field Level Validation Rules

Electricity transportation contracts which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

7.3.1 Field No 1 / No 2: Document identification / Document version

Rule 1: If the pair (Document identification, Document version) contained in the submitted document matches an existing pair of an already submitted document, then the whole document is rejected.

Rule AT3F1R1

If (document ->mRID, document->revisionNumber) matches an existing record with same values for mRID and revisionNumber, then the document is rejected.

- *Error code: AT3F1E1*
- *Error message: The submitted document is duplicated*

7.3.2 Field No 4 Sender Identification

Rule 1: The ID of the Sender (*sender_MarketParticipant.mRID*) must have a match in the list of RRM's; furthermore the ID of the Sender must correspond to the ACER code used in the name of the submitted file.

For example, if the submitted filename is 20150715_ElectricityRights_V1_A0000116N.UK_1.xml.asc.pgp, then the ID of the Sender must correspond to the ACER code **A0000116N.UK**.

Rule ReportingEntityTRANS

If ReportingEntityID is NOT in the list of RRM's

OR

*ReportingEntityID does not correspond to the ACER code used for submitting the file,
then the contract report is invalid*

- *Error code: R1REPENTCON*
- *Error message: Invalid Id of Reporting Entity*

Rule 2: If the Sender Identification does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT3F4R1

If document -> sender_MarketParticipant.mRID does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern "[0-9][0-9][XYZTWV].+"

then the document is rejected.

- *Error code: AT3F4E1*
- *Error message: Invalid EIC code format*

7.3.3 Field No 6 Receiver Identification

Rule 1: If the Receiver Identification does not match ACER EIC code, then the whole document is rejected.

Rule AT3F6R1

If document -> receiver_MarketParticipant.mRID does not match ACER EIC code, then the document is rejected.

- *Error code: AT3F6E1*
- *Error message: Invalid ACER EIC code*

7.3.4 Field No 10 Domain

Rule 1: If the Domain does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT3F10R1

If document ->domain.mRID does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT3F10E1
- Error message: Invalid EIC code format

7.3.5 Field No 19 In Area

Rule 1: If the In Area does not match a predefined patter for EIC code compliance than the whole document is rejected.

Rule **AT3F19R1**

If document ->TimeSeries -> in_Domain.mRID does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT3F19E1
- Error message: Invalid EIC code format

7.3.6 Field No 20 Out Area

Rule 1: If the Out Area does not match a predefined patter for EIC code compliance than the whole document is rejected.

Rule **AT3F20R1**

If document ->TimeSeries -> out_Domain.mRID does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT3F20E1
- Error message: Invalid EIC code format

7.3.7 Back loading of historical data for Phase 2 reporting

The following rules apply for Electricity Transportation Contracts:

Rule 1: the system will allow back loading of historical data skipping any validation rule.

Rule AT3F8BLR1

If the date in the filename of the submitted XML file is prior (<) to the Date1 (5 Oct 2015 00:00:00Z), then all validation rules (except the ones defined in this section) will be ignored.

If the document in the submitted XML file with the date in the filename prior (<) to the Date1 (5 Oct 2015 00:00:00Z) has a **createdDateTime** >= the Date3 (7 Apr 2016 00:00:00Z), then the document is rejected;

- *Error code: AT3F8BLE1*
- *Error message: Expected back loading data received at least one live data*

Rule AT3F8BLR2

If the date in the filename of the submitted XML file is greater or equal to Date1 (5 Oct 2015 00:00:00Z) and the document has a **createdDateTime** < Date4 (4 Apr 2016 00:00:00Z), then document will be rejected.

- *Error code: AT3F8BLE2*
- *Error message: Expected live data received at least one back loading data*

Note: Backloading has been discontinued since 30 June 2017

7.4 Annex Table 4 Field Level Validation Rules

Gas transportation contracts which fail validation will be handled in accordance with the rules for handling validation failure in section 3.

7.4.1 Document identification / Document version

Rule 1: If the pair (Document identification, Document version) contained in the submitted document matches an existing pair of an already submitted document, then the whole document is rejected.

Rule AT4U1R1

If (document ->identification, document->version) matches an existing record with same values for "identification" and "version", then the document is rejected.

- *Error code: AT4U1E1*

- *Error message: The submitted document is duplicated*

7.4.2 Recipient Identification

Rule 1: If the Recipient Identification does not match ACER EIC code, then the whole document is rejected.

Rule AT4U2R1

If document -> recipient_MarketParticipant.identification does not match ACER EIC code, then the document is rejected.

- *Error code: AT4U2E1*
- *Error message: Invalid ACER EIC code*

7.4.3 Field No 1 Issuer Identification

Rule 1: The ID of the Issuer (*issuer_MarketParticipant.identification*) must have a match in the list of RRM; furthermore the ID of the Issuer must correspond to the ACER code used in the name of the submitted file.

For example, if the submitted filename is 20150715_GasCapacity_V1_ **A0000116N.UK**_1.xml.asc.pgp, then the ID of the Issuer must correspond to the ACER code **A0000116N.UK**.

Rule ReportingEntityTRANS

If ReportingEntityID is NOT in the list of RRM

OR

ReportingEntityID does not correspond to the ACER code used for submitting the file,

then the contract report is invalid

- *Error code: R1REPENTCON*
- *Error message: Invalid Id of Reporting Entity*

Rule 2: If the Issuer Identification does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT4F1R1

If document -> issuer_MarketParticipant.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT4F1E1
- Error message: Invalid EIC code format

7.4.4 Field No 2 Organised Market Place Identification

Rule 1: If the Organised Market Place Identification does not match a predefined patter for EIC code compliance than the whole document is rejected.

Rule **AT4F2R1**

If document -> organisedMarketPlace_MarketParticipant.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT4F2E1
- Error message: Invalid EIC code format

7.4.5 Field No 22 Connection point Identification

Rule 1: If the connection point identification does not match a predefined patter for EIC code compliance than the whole document is rejected.

Rule **AT4F2R1**

If document -> process_Transaction.connectionPoint.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT4F22E1
- Error message: Invalid EIC code format

7.4.6 Field No 25 Responsible TSO Identification

Rule 1: If the Responsible TSO identification does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT4F25R1

If document -> process_Transaction.responsibleTso_MarketParticipant.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern "[0-9][0-9][XYZTWV].+"

then the document is rejected.

- *Error code: AT4F25E1*
- *Error message: Invalid EIC code format*

7.4.7 Field No 26 Adjacent TSO Identification

Rule 1: If the Adjacent TSO identification does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT4F26R1

If document -> process_Transaction.adjacentTso_MarketParticipant.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern "[0-9][0-9][XYZTWV].+"

then the document is rejected.

- *Error code: AT4F26E1*
- *Error message: Invalid EIC code format*

7.4.8 Field No 27 Primary Market Participant Identification

Rule 1: If the Primary Market Participant identification does not match a predefined pattern for EIC code compliance then the whole document is rejected.

Rule AT4F27R1

If document -> Transportation_Transaction -> primary_MarketParticipant.identification does not match the following restrictions:

Maximum length 16 characters

Regular expression pattern “[0-9][0-9][XYZTWV].+”

then the document is rejected.

- Error code: AT4F27E1
- Error message: Invalid EIC code format

7.4.9 Back loading of historical data for Phase 2 reporting

The following rules apply for Gas Transportation Contracts:

Rule 1: the system will allow back loading of historical data skipping any validation rule.

Rule **AT4F6BLR1**

If the date in the filename of the submitted XML file is prior (<) to the Date1 (5 Oct 2015 00:00:00Z), then all validation rules (except the ones defined in this section) will be ignored.

If the document in the submitted XML file with the date in the filename prior (<) to the Date1 (5 Oct 2015 00:00:00Z) has a **createdDateTime** >= the Date3 (7 Apr 2016 00:00:00Z), then the document is rejected;

- Error code: AT4F6BLE1
- Error message: Expected back loading data received at least one live data

Rule **AT4F6BLR2**

If the date in the filename of the submitted XML file is greater or equal to Date1 (5 Oct 2015 00:00:00Z) and the document has a **createdDateTime** < Date4 (4 Apr 2016 00:00:00Z), then document will be rejected.

- Error code: AT4F6BLE2
- Error message: Expected live data received at least one back loading data

Note: Backloading has been discontinued since 30 June 2017

8 Fundamental Data Functional Data Validation

Fundamental data has several unique types, but has several underlying requirements for the representation and understanding of the data.

8.1 Field Integrity Compliance

8.1.1 Participant Code Validation

In this validation, any field that identifies a registered participant with ACER for example the owner of a power plant will be validated against the real values within the database to ensure that the values reported actually exist. If a participant field is incorrectly reported, then the transaction needs to be highlighted to ensure that the participant is correctly identified.

8.1.2 Time Period Validation

All fundamental data must clearly identify the time period for which it is applicable. This is defined in the time period fields within the data set, identifying when the start and end of the time period is.

The defined start period must be prior to the defined end period and the defined period must define a single contiguous period of time.

8.2 Functional Data Quality Compliance

8.2.1 Geographical Location Representation Validation

The primary key for all fundamental data is the geographical representation of the data. For example, all data is relevant only to a specific zone or region of Europe. All fundamental data will be validated to ensure that the geographical data relating to the fundamental data has been correctly represented.

For example, if a product is represented using an EIC code, this has an associated location for the delivery of the product. This must be represented as a geographical coordinate, i.e. latitude and longitude.

8.2.2 Fundamental Data Type Validation

The fundamental data formats have been defined by the ENTSO-E and ENTSO-G organisations. The data type validation is therefore assured to be accurate based on the definition of the schemas provided by these organisations. ACER does not have responsibility for these data formats and therefore can't be responsible for validating that the data types or data formats are valid.

Any inaccurately reported data will be assessed by ACER and notification will be passed to ENTSO-E and ENTSO-G as the responsible organisations to ensure that the data quality is achieved.

8.3 Fundamental Field Data Validation

8.3.1 Valid EIC Code Format

Where a schema indicates the use of a EIC code, the EIC code must comply with the rules for EIC code construction as specified by ENTSO-E and ENTSO-G. ACER will validate that the format of the EIC code complies and will notify ENTSO-E and ENTSO-G if any transactions with incorrectly formatted EIC codes are received.

8.3.2 Valid EIC Codes Received

Where a schema references a participant or area via EIC code, the EIC code must be registered with ENTSO-E or ENTSO-G. If the EIC code is identified as not registered with ENTSO-E or ENTSO-G, ACER will notify ENTSO-E and ENTSO-G that transactions with invalid or unregistered EIC codes have been received.

8.3.3 No Duplicated Time Series

When a fundamental data report is received including time series events, any duplicated time series for the same event will be identified. If a duplicate time series is identified, then ACER will notify ENTSO-E and ENTSO-G of the duplicate transaction.

Note: A duplicate time series is defined per fundamental data reporting schema, but requires all values within a time series to be duplicated with the exception of the reported data quantity field.

8.4 Fundamental Field Level Validation Rules

8.4.1 Field: Reporting entity ID (sender_MarketParticipant.mRID / issuer_MarketParticipant.identification / reportingEntityIdentifier)

Rule 1: The ID of the Reporting Entity (sender_MarketParticipant.mRID for Electricity, issuer_MarketParticipant.identification for Gas, reportingEntityIdentifier for LNG and Storage) must have a match in the list of RRM; furthermore the ID of the Reporting Entity must correspond to the ACER code used in the name of the submitted file.

For example, if the submitted filename is 20150715_ElectricityConfiguration_V1_A0000116N.UK_1.xml.asc.pgp, then the ID of the Reporting entity must correspond to the ACER code **A0000116N.UK**.

Rule ReportingEntityFND

*If ReportingEntityID is NOT in the list of RRM*s

OR

*ReportingEntityID does not correspond to the ACER code used for submitting the file,
then the contract report is invalid*

- *Error code: R1REPENTCON*
- *Error message: Invalid Id of Reporting Entity*

9 Data Correction Validation

This section defines additional validation checks that will be performed when data is submitted as correction to existing data. These validations only apply if a RRM has received notification that some of the data that they reported are invalid and must be corrected.

9.1 Correcting an Existing Transaction Validation

The system will store a list of all transactions that have been reported as invalid back to each of the participants. When a participant submits a correction report, this will be validated against the list of transactions that need to be corrected and it will be ensured that the transaction details that need to be maintained as consistent for the transaction are managed correctly.

For example, a transaction given a specific reference identification number applies to a unique set of keys for the transaction, such as the product identification, the transaction type etc. If these values don't match with the original transaction reference values, then a new transaction should be created and an error report for the original transaction should be provided so that the original transaction is identified as erroneous.

Each transaction which has failed any of the validation steps in the sections Technical Data Validation or Standard Contract Functional Data Validation will be identified as a transaction that needs to be corrected.

A participant must retrieve the receipt of transactions and identify that the transactions have failed validation and then resubmit the transactions to the system. Each transaction which has failed will be identified by a unique transaction reference number, which the participant will need to maintain to ensure that the correction is correctly identified.

The output schema contains a reference to the transaction, which is the original transaction reference identifier, but also to the assigned transaction reference for tracking the correction process.

Once a correction report is received from the market participant, each of the transactions which are due for correction by that market participant are validated against the updated correction report. If a transaction is received that is not in the existing transaction correction report, it is identified not as a correction, but as a late-reported transaction and is subject to the addition checks as defined in section Late Report Transaction Validation.

10 Late Report Transaction Validation

This section defines the additional checks, on top of those performed for technical data validation and functional validation, for transactions which are reported late. This applies to any transactional report that is submitted for a date outside of the reporting window for reporting transactions.

10.1 Duplicate Reporting Transaction Validation

When a participant reports transactions late, each one is validated against existing transactions. This is similar to the normal duplicate transaction report, both within the data that has been submitted by the participant, but also it is validated against every transaction that has been submitted by the participant on the correct date of reporting. All of the normal duplicate transaction validations occur, i.e. validation against unique fields, but in the case of late reported transactions it also validates against different states of the lifecycle of the transaction to ensure that the transaction lifecycles have been adhered to.

10.2 Transaction Linking Validation

All transactions should have a valid linking between orders and trades, for example if a trade has been reported correctly at the correct time, but the order is reported late, there is an additional validation that any linking identifier supplied within the transactions are correct, i.e. the linking transaction identifier code (i.e. the code that links the order to trade or order to order etc.) is actually correct and matches an existing record. If a record is reported late with an unknown linking transaction identifier, this is reported as an error, however the transaction can be accepted, but the missing linking transaction would be reported as an invalid transaction or transaction that has not been received.

Annex I: Error conditions for Data Exchange Process

Some error conditions can arise while interacting with the system; the following paragraphs take under consideration each tier where specific errors can occur.

Validation Status and Error messages

During the processing of a file, some errors can arise. They are related to the status they reached during the validation phase and description messages shown are therefore dynamically evaluated based on the status description. The following list shows all status messages and codes available in the final receipt generated by ARIS:

| Status Id | Status Short name | Status full description | Level | Notes |
|-----------|-----------------------------------|--|-------------------------|---|
| 23 | Tier2 Success | Tier2: Data validated successfully | Tier2 | Data fully processed in tier2 |
| 89 | Tier1-2 Integration Failure | Tier1-Tier2: Failed to propagate data | Tier1/Tier2 Integration | low level error in data transfer, entails a system failure and investigation |
| 90 | Party Id Failure | Failure referring to invalid or not existent Party Id | Tier1 | RRM Id not valid (non-existent) |
| 91 | Zip not accepted Failure | Failure due to file being a .zip Archive | Tier1 | Upload file is a zip (only sftp) |
| 92 | Invalid Extension Failure | Failure due to file having a generic unaccepted file extension | Tier1 | file is not .xml.asc.pgp |
| 93 | Naming Convention Generic Failure | Generic exception during naming convention verification | Tier1 | One of the parts contained a completely nonsensical information (e.g. where the version of the schema is supposed to be provided, you provide a letter) |
| 94 | Invalid Date Failure | Failure proceeding due to invalid date in filename | Tier1 | Invalid filename date (relates to submission date to be today) |

| Status Id | Status Short name | Status full description | Level | Notes |
|-----------|-------------------------------------|---|-------|---|
| 95 | Invalid Schema Reference Failure | Reference to a schema that does not exist or is not allowed | Tier1 | reference to a not allowed or non-existent schema in the filename of the submission |
| 96 | Invalid Schema Validation Failure | Failure during the Schema Validation Phase | Tier1 | XML to XSD schema validation failure |
| 97 | Decryption or Decompression Failure | Failure during the Decompression & Decryption Phase | Tier1 | decompression//decryption phase error |
| 98 | Naming Constraint Failure | Failure proceeding due to naming convention Constraint | Tier1 | filename structure was incorrect (too few or too many "part") |
| 99 | Workflow Transition Failure | Failure proceeding to next workflow step | Tier1 | low level error, needs to be investigated |
| 101 | Invalid SEQID Failure | Failure missing or invalid SEQID in the filename | Tier1 | SEQ Id inconsistent |
| 102 | Invalid Duplicated Filename Failure | Failure specified filename is already present! | Tier1 | filename duplicated |
| 103 | Disabled User | Upload Refused: related user disabled | Tier1 | upload made via a disabled user (sftp) |
| 104 | Disabled Entity | Upload Refused: related Reporting Entity is disabled | Tier1 | upload relevant to a disabled RRM |
| 105 | Unmatching Identity | Upload Refused: Reporting Entity in filename and Reporting Entity associated with identity are different! | Tier1 | uploader RRM and filename RRM do not match |

| Status Id | Status Short name | Status full description | Level | Notes |
|-----------|--------------------------------------|---|-------------------------|---|
| 106 | PGP Signature Failure | Failure during the PGP signature verification Phase | Tier1 | the PGP signature was missing, invalid or inconsistent |
| 113 | Metadata Error | Metadata referring to wrong Reporting Entity | Tier1/Tier2 Integration | the metadata inside the file points to a different RRM from the uploader |
| 122 | Tier2 Partial Success | Tier2: Partial Success for Validation Rules | Tier2 | Data was processed, but at least some of it raised an error due to validation rules |
| 123 | Duplicated Identifier | Part of the reported data referenced a duplicated identifier (EIC, LEI, BIC or GLN) | Tier2 | The file is rejected since it contains references to duplicated identifier |
| 124 | Tier2 Failure / Unknown System Error | Tier2: Failure | Tier2 | Low level error in content elaboration, entails a system failure; please report the error to the CSD with full information about RRM user, load ID, interface type. |

The following status messages can be displayed on the DCI interface (Web GUI and Web Services) but they will not appear on the generated receipt file since they are related to internal transition phases:

| Status Id | Status Short name | Status full description | Level | Notes |
|-----------|-------------------|-------------------------|-------|---------------------------------|
| 0 | Inserting | Inserting | Tier1 | file submitted |
| 11 | Process running | Process Running | Tier1 | Scheduler starts the processing |
| 13 | Upload | Tier1: Upload Done | Tier1 | file submitted (sftp) |

| | | | | |
|----|------------------|--|-------------------------|--|
| 14 | Naming | Tier1: Naming Convention Validation Done | Tier1 | first phase of control |
| 15 | File Elaboration | Tier: Decompression & Decryption Done | Tier1 | decryption, decompression , signature validation |
| 16 | XSD Validation | Tier1: Schema Validation Done | Tier1 | schema validation |
| 17 | Tier1 On Hold | Tier1: Waiting for Tier2 Support | Tier1 | tier2 does not yet support submitted file |
| 18 | Tier1 Done | Tier1: Available for Tier2 | Tier1 | Tier1 elaboration completed |
| 19 | Tier2 Phase1 | Tier2: Elaboration Phase1 | Tier2 | Tier2 coherency (optional) |
| 20 | Tier2 Phase2 | Tier2: Elaboration Phase2 | Tier2 | Tier2 validation (mandatory) |
| 21 | Tier2 to DWH | Tier2: Data moved to DWH for validation | Tier1/Tier2 Integration | Tier1 to tier2 data move |