

Reporting Instructions

MIDAS System

Document Control Information

Document Title:	Reporting Instructions Document
Project Title:	MIDAS
Project Owner:	Fabrizio Planta
Solution Provider:	European Dynamics
Project Manager:	Mansoor Ullah
Doc. Version:	V1
Date:	02/09/2026

Document History

Version	Date	Author	Section / Description of Change
1	02/09/2026	ESMA	Version 1 for publication

Reference Documents

Ref	Document Title	Version	Author	Date
1	ESMA65-955014868-15160 MiCA Order Book and Records Keeping Message Specifications		ESMA	28 Nov 2025

Table of Contents

1	Introduction	8
1.1	Background.....	8
1.2	Scope	9
1.3	Definitions	10
2	System overview.....	11
3	Main principles	12
3.1	Reporting principles	12
3.2	Files specifications.....	12
3.3	File naming convention	12
3.4	Data collection	14
3.4.1	Business fields.....	14
3.4.2	Additional Mandatory fields at JSON syntax level	40
3.5	Data validation	48
3.5.1	Data transmission/file validation rules	48
3.5.2	Data content validation rules.....	48
3.5.3	Feedback files.....	51
3.5.4	Error correction process.....	52
4	Onboarding and Upload specifications.....	54
4.1	Onboarding Process	54
4.1.1	Establish Access.....	54
4.1.2	Prepare JSON file	55
4.1.3	Upload of test sample JSON file(s) on UAT	56
4.1.4	Release to Production.....	56
4.2	Communications	56
4.3	Support	56
5	Annex 1: Field type definitions	57
6	Annex 2: File format: JSON schema	58
7	Annex 3: Data validations	59
7.1	Transmission/file Validation Rules	59

7.2	Content Validation Rules.....	59
7.2.1	Rejection validation rules	59
7.2.2	Qualitative validation rules	62
8	Annex 4: JSON file types	64
9	Annex 5: Feedback file.....	65
9.1	Purpose and scope	65
9.1.1	What the feedback file is	65
9.1.2	1.2 What the feedback file is not	65
9.1.3	1.3 Scope	65
9.2	Delivery.....	66
9.2.1	When the feedback file is generated	66
9.2.2	Where the feedback file is written	66
9.2.3	Relationship between submitted files and feedback files.....	66
9.2.4	Naming convention	66
9.2.5	Retention	67
9.2.6	Format	67
9.3	How to read the file	67
9.3.1	Format	67
9.3.2	The governing principle: one row per finding, not one row per record	67
9.3.3	The four kinds of row	68
9.3.4	Reading order	68
9.4	Column dictionary	69
9.4.1	ReportTimestamp	69
9.4.2	CaspIdentifier.....	70
9.4.3	SubmittedFileName	70
9.4.4	FeedbackType	70
9.4.5	FileVerdict.....	70
9.4.6	RecordRef	71
9.4.7	JSONLineNumber.....	71
9.4.8	Outcome	72

9.4.9	ErrorCode	72
9.4.10	AppliesTo	72
9.4.11	Label	72
9.4.12	Severity	73
9.4.13	Field	73
9.4.14	LogicalPath	74
9.4.15	SubmittedValue	74
9.4.16	ValidationReason	75
9.4.17	CorrectiveAction	75
9.5	Attribute values	75
9.5.1	FeedbackType	75
9.5.2	FileVerdict	75
9.5.3	Outcome	76
9.5.4	Severity	76
9.5.5	AppliesTo	76
9.5.6	ErrorCode and Label	77
9.5.7	CorrectiveAction patterns	77
9.6	Legal combinations	78
9.7	Conventions	79
9.7.1	Empty, -, and literal values	79
9.7.2	Timestamps	79
9.7.3	Composite rules	79
9.7.4	Row order	80
9.7.5	Stability	80
9.8	Worked walkthrough of the sample	80
9.8.1	A file rejected before processing	80
9.8.2	A processed file with rejected records	81
9.8.3	A record accepted but carrying a warning	81
9.8.4	A whole-file quality warning with no record reference	81
9.8.5	An accepted record	82

9.8.6	One record breaching several rules	82
9.8.7	Locating a record where RecordRef repeats	83
9.8.8	Coverage of the sample	83
9.9	Acting on the feedback file	83
9.9.1	Deriving the resubmission set	83
9.9.2	Automated handling	84

Table of tables

Table 1 - Definitions	10
Diagram 2 - Data Flowchart	11
Table 3 - Executions field attributes	15
Table 4 - Trading date time	16
Table 5 - Transaction Record Number	17
Table 6 - Trading platform for crypto-asset transaction identification code	18
Table 7 - Price Currency	18
Table 8 - Crypto Asset Identification Code	19
Table 9 - Side	20
Table 10 - Quantity	21
Table 11 - Price	22
Table 12 - Trading platform for crypto-asset	23
Table 13 - Orderbook data attributes	23
Table 14 - Date and time	24
Table 15 - New order, order modification, order cancellation, order rejections, partial or full execution	26
Table 16 - Order identification code	27
Table 17 - Transaction identification code	28
Table 18 - Order type classification	29
Table 19 - Buy-sell indicator	29
Table 20 - Initial quantity	30
Table 21 - Price currency	31
Table 22 - Crypto-asset identification code	32
Table 23 - Remaining quantity	32
Table 24 - OrderPassiveOnly	33
Table 25 - Traded quantity	34
Table 26 - Limit price	35

Table 27 - Passive or aggressive indicator	36
Table 28 - Validity period	37
Table 29 - Validity period and time	38
Table 30 - Segment MIC	39
Table 31 - Transaction price	40
Table 32 - Seller Branch Indicator	40
Table 33 - Buyer Branch Indicator	41
Table 34 - Report Header Reporting Entity LEI	42
Table 35 - Report Header Date	42
Table 36 - Report Identification	42
Table 37 - Order Book Identification	43
Table 38 - Sequence number	43
Table 39 - Client Exception Identification	44
Table 40 - Non Executing Broker Party Exception	44
Table 41 - Trading Capacity	45
Table 42 - Stop Price (No Price - Pending)	46
Table 43 - Additional Limit Price (No Price – Pending)	46
Table 44 - Pegged Price (No Price - Pending)	47
Table 45 - Field type definitions	57
Table 46 - Transmission/file validation rules	59
Table 47 - Rejection validation rules	61
Table 48 - Qualitative validation rules	63

1 Introduction

1.1 Background

This document specifies the reporting instructions for the provision of data necessary to support the integrity of crypto-asset markets, including market abuse supervision under Title VI of MiCA. The primary purpose of these instructions is to define the file formats, submission process, naming convention, validation rules, and data content required from Crypto-Asset Service Providers (CASPs) to provide to the MiCA Digital Assets Surveillance system (MIDAS).

MIDAS is the system designed to ingest data from CASPs to be used for the monitoring of market abuse and to support overall market integrity under MiCA. It is based on the following legal provisions:

1. The Regulation (EU) 2023/1114 on Markets in Crypto-Assets (MiCA), came into force in June 2023, with the provisions applicable to CASPs applying from 30 December 2024. Within its provisions, Article 94 of MiCA, provides that in order to perform its duties under Titles II to VI of the same Regulation, National Competent Authorities (NCAs) are granted relevant powers, including the power to require any person to provide information and documents that they consider could be relevant for the performance of their duties. This provision is to be read in conjunction with the provisions regarding the record keeping obligation under MiCA Articles 68(9) and 76(15), which are functional to NCAs' fulfilment of their supervisory and market integrity tasks.
2. Article 68(9) of MiCA holds that CASPs shall keep records of all crypto-asset services, activities, orders, and transactions. The detailed record-keeping requirements are specified in the RTS 2025/1140, which defines the records to be maintained, including a sub-set of on-chain data elements. These records shall be sufficient to enable competent authorities to fulfil their supervisory tasks and to take enforcement measures, and in particular to ascertain whether CASPs have complied with all obligations including those with respect to clients or prospective clients and to the integrity of the market.
3. Moreover, in accordance with Article 76(15) of MiCA, CASPs operating a trading platform for crypto-assets shall keep at the disposal of NCAs, for at least five years, the relevant data relating to all orders in crypto-assets that are advertised through the systems of CASPs operating a trading platform, or give access to the NCA, to monitor the trading activity. This data also includes a sub-set of on-chain data elements.

ESMA has no formal supervisory role in MiCA. MiCA assigns supervisory and enforcement responsibilities to NCAs. Therefore, NCAs remain responsible for day-to-day market surveillance, investigations and enforcement. They are the authorities that decide whether to follow up on an

alert or a data quality issue, request additional information, launch an investigation, or impose sanctions. However, with MIDAS, the NCAs collectively decided to centralise the technical capability needed to collect and analyse data on crypto-assets markets, which are inherently cross-border and technologically complex and require a common analytical foundation to monitor the markets at EU-level as no single authority can effectively do that in isolation at local level. This decision has been implemented via a “Delegation Agreement”, which is a tool envisaged under Article 28 of ESMA regulation¹ that allows NCAs to delegate some of their tasks to ESMA. A delegation can only cover tasks and not supervisory responsibilities. Under the delegated project, NCAs have delegated to ESMA the tasks of collecting data, processing it, performing quality checks, calculating common indicators of suspicious Market Abuse behaviours (alerts) and distributing those alerts to the relevant NCAs. Following a public procurement process, ESMA has contracted a third-party specialised provider for this delegated project.

For MIDAS, the data collection framework does not mandate the transmission of the full set of records subject to the record-keeping obligations under Articles 68(9) and 76(15) of MiCA. Instead, under the Delegation Agreements executed between NCAs and ESMA, MIDAS operates exclusively on a defined subset of data elements necessary for market monitoring and the detection of market abuse. This subset was subsequently extended by the Delegated Project Board to include additional pre-trade data fields, specifically order side, buy or sell indicators, and order quantities including cancellations. This targeted expansion aims to improve alert calibration and minimize false positives, while ensuring that data collection remains limited to what is strictly necessary for supervisory purposes. The precise scope of these requirements is implemented in phases, starting with a smaller subset of data and gradually increasing the scope as supervisors build knowledge and experience in crypto markets, and as CASPs build their implementation capacity.

1.2 Scope

The scope of this document covers the reporting instructions for the submission of a defined subset of data to the MIDAS tool. In alignment with the Delegation Agreements executed with ESMA, this data collection is divided into two distinct functional categories based on the reporting entity's regulatory obligations under MiCA:

1. Executions Data (Record-Keeping Data auth.117): This category covers executed transactions. Submission of this data subset is **mandatory for all CASPs** operating within the EU regulatory framework under the record-keeping obligations of MiCA Article 68(9) and are specified in the RTS 2025/1140.
2. Orderbook Data (auth.118): This category covers the pre-trade order lifecycle data. Submission of this data subset is mandatory only for **CASPs operating a trading**

¹ [EUR-Lex - 02010R1095-20251110 - EN - EUR-Lex](#)

platform for crypto-assets under MiCA Article 76(15) and are specified in the RTS 2025/416.

Record keeping is mandatory for all CASPs operating within the EU regulatory framework².

1.3 Definitions

Acronym	Definition
CASP	Crypto-Asset Service Provider
DTI	Digital Token Identifier
ESMA	European Securities and Markets Authority
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LEI	Legal Entity Identifier
MIC	Market Identifier Code
RTS	Regulatory Technical Standards
SFTP	Secure File Transfer Protocol
UAT	User Acceptance Testing

Table 1 - Definitions

² For the purposes of these instructions, the term 'reporting' is used to describe the submission of data in response to requests from competent authorities, in accordance with the RTS message specifications and technical schemas. While MiCA establishes record-keeping obligations and RTS 2025/1140 further specifies the records to be kept, the RTS also envisages the provision of records to NCAs using harmonised standards and refers to consistent reporting standards across the financial sector.

2 System overview

The MIDAS system acts as a central hub for crypto-asset market surveillance. CASPs are responsible for implementing submission processes to provide data in the specified JSON format.

1. **Submission:** CASPs upload JSON files to a dedicated SFTP drop point.
2. **Validation:** Files undergo three levels of check: Transmission, File and Content validations.
3. **Alerting:** The system applies detection algorithms to identify potential market abuse and generate alerts for NCAs.

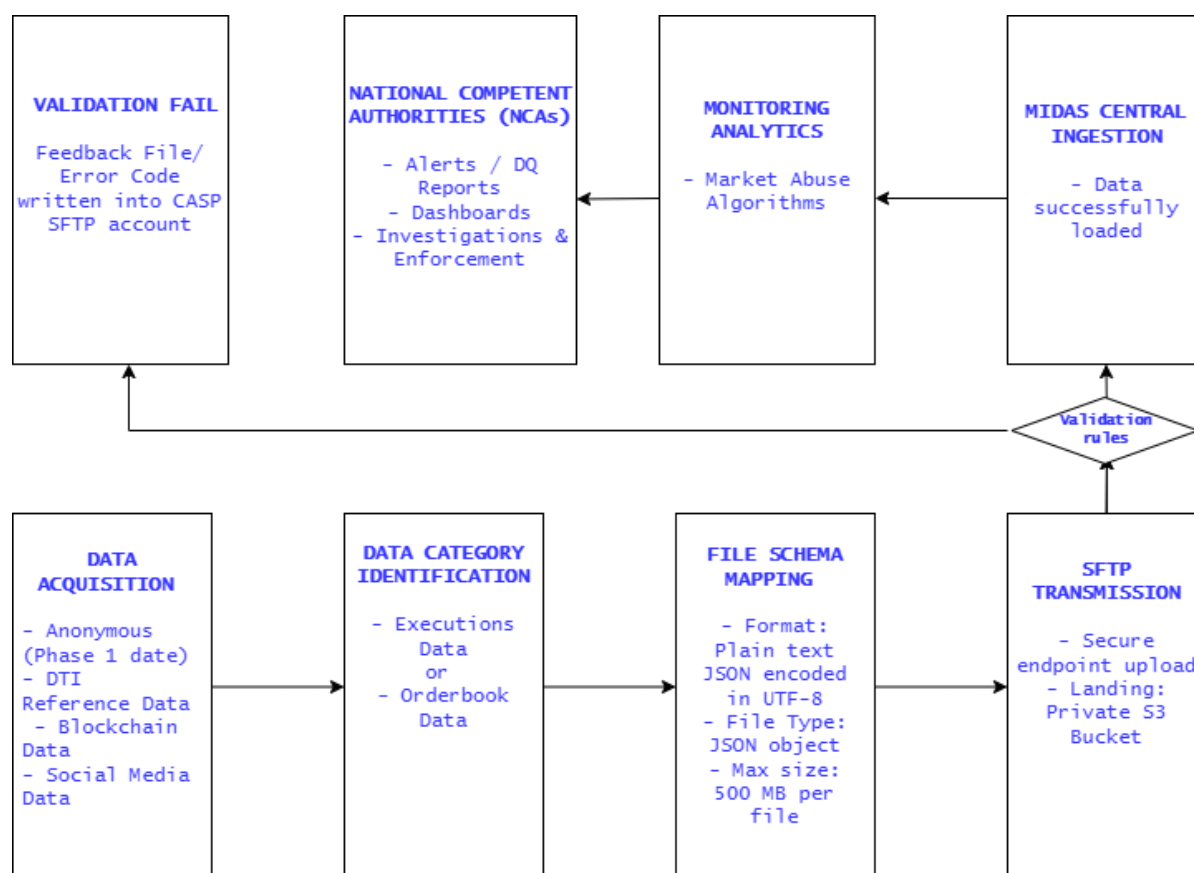


Diagram 2 - Data Flowchart

3 Main principles

3.1 Reporting principles

MIDAS follows a phased implementation approach aimed at limiting implementation costs, complexity and operational risk. Under its first phase, MIDAS will collect only a limited set of on- and off-chain data elements necessary for EU-level market abuse monitoring and alert generation³. Consequently, MIDAS does not collect the full set of records subject to the record-keeping obligations under Articles 68(9) and 76(15) of MiCA. In line with its phased implementation approach, the first phase excludes personal data, non-public CASP transaction and supervisory data, as well as information collected for enforcement or investigative purposes (such as STORs and RFIs). Future implementation phases will extend the scope of MIDAS to include supervisory data, including personal identifiers, where required for supervisory purposes.

CASPs operating a trading platform must submit both record keeping executions data (auth.117) and pre trade orderbook data (auth.118). In record keeping executions both buy and sell legs must be included in the same record (if applicable).

All other CASPS must only submit record keeping executions data (auth.117). Only relevant buy or sell leg must be included in one record.

3.2 Files specifications

All reports must be submitted exclusively as JSON files conforming to the order book and record keeping JSON files published on MiCA and provided in

³ In line with this principle, for execution messages, buyer and seller identification elements shall not be reported for the purpose of deriving the Side value. Instead, the Side value shall be populated using the applicable default value.

Annex 2: File format: JSON schema.

- **Format:** Plain text JSON encoded in UTF-8.
- **Exclusivity:** Each file may contain at least one record.
- **Size Limit:** Each individual JSON file must not exceed 500 MB. Larger datasets must be split into multiple parts.
- **Completeness:** Submitting entities must pre-validate files against the provided JSON schema to ensure all mandatory fields are present and provided as per the defined datatype.

3.3 File naming convention

To facilitate automated ingestion, MIDAS requires a strict naming convention and complete submission of the data. Files that deviate from this format will be automatically rejected.

Naming convention: <FileType>_<DateDDMMYYYY>_<partN>_<partM>.json

- **FileType:** Specifies the message type. Possible FileType values are *CryptoAssetRecordKeeping* and *CryptoAssetOrderBook*.
- **DateDDMMYYYY:** The record keeping date the data refers to (e.g., 15052026).
- **partN:** For data larger than 500 MB, the file must be split into multiple parts to stay under the size limit. To preserve the order, a sequential suffix for multi-part submissions is introduced and it must increase sequentially (e.g., _1, _2). No periods or special characters other than underscore should precede the suffix of the filename(s). The minimum value of the suffix must be 1 and there is no upper limit in the maximum number. The partN numbers must be reset for each <DateDDMMYYYY>.
- **partM:** Suffix denoting the total number of split files for a Date and Filetype. For data smaller than 500 MB, the suffix will be 1 as the dataset only consists of one file. No periods or special characters other than underscore should precede the suffix of the filename(s). The minimum value of the suffix must be 1 and there is no upper limit. The partM number must be reset for each combination of <FileType> and <DateDDMMYYYY> as it always represents the total number of files for this specific submission.

Examples:

Execution (Record Keeping):

- Data smaller than 500 MB:
 - *CryptoAssetRecordKeeping_12042026_1_1.json*
- Data larger than 500 MB, i.e., 1500 MB:

- CryptoAssetRecordKeeping_15052026_1_3.json
- CryptoAssetRecordKeeping_15052026_2_3.json
- CryptoAssetRecordKeeping_15052026_3_3.json
- Incomplete submission (files two and four missing):
 - CryptoAssetRecordKeeping_24052026_1_4.json
 - CryptoAssetRecordKeeping_24052026_3_4.json

OrderBook:

- Data smaller than 500 MB:
 - CryptoAssetOrderBook_12042026_1_1.json
- Data larger than 500 MB but smaller than 1500 MB:
 - CryptoAssetOrderBook_15052026_1_3.json
 - CryptoAssetOrderBook_15052026_2_3.json
 - CryptoAssetOrderBook_15052026_3_3.json
- Incomplete submission (file two missing):
 - CryptoAssetOrderBook_01052026_1_3.json
 - CryptoAssetOrderBook_01052026_3_3.json

File names are validated by the MIDAS system upon upload. Files not following the required naming convention (e.g. wrong date format, missing part number where applicable, incorrect file extension, or unrecognized FileType) will be rejected as part of the validation process.

3.4 Data collection

CASPs are required to submit their data files every two weeks to a designated SFTP location mapped directly to the MIDAS system. Should a CASP prefer to submit data at a different frequency, this can be accommodated upon agreement with the MIDAS provider and the NCAs. All submissions must be provided exclusively in plain text JSON format. All files submitted in any other formats will be automatically rejected by the system.

3.4.1 Business fields

3.4.1.1 Overview

This section defines the initial set of data which will be sent to ESMA (MIDAS System) separated in the following two categories based on the record-keeping entity's regulatory obligations under MiCA:

1. Executions (Record Keeping) business fields (auth.117)
2. Orderbook business fields (auth.118)

3.4.1.2 Description of execution business fields

Each execution business field is structured into standard attributes, as outlined in the table below:

List of Executions Fields Attributes	Attribute Description
Example	Execution field example
RTS 2025/1140 Annex – Table 3 Field	Field number according to RTS 2025/1140 Annex – Table 3
RTS 2025/1140 Annex – Table 3 Field Name	Field name according to RTS 2025/1140 Annex – Table 3
Logical Path	Logical location on JSON schema to report/distribute this element
Type / Code	Data type
Field Description	Description of the field
Required Field	Mandatory / Optional
Validation Rules	Specific validation rules to be applied to this field

Table 3 - Executions field attributes

3.4.1.3 List of execution fields

1. Trading date time

Trading date time	Description
Example	2026-03-15 14:25:36.125
RTS 2025/1140 Annex – Table 3 Field	[3.27]
RTS 2025/1140 Annex – Table 3 Field Name	Trading date time
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/Tx/TradDt
Type / Code	dateTime
Field Description	Date and time when the transaction was executed (UTC time). For transactions not executed on a trading venue, the date and time shall be when the parties agree to the content of the following fields: quantity, price, currencies, instrument identification code, instrument classification and underlying instrument code, where applicable. For transactions not executed on a trading venue, the time recorded shall be at least to the nearest second. Where the transaction results from an order transmitted by the executing firm on behalf of a client to a third party where the conditions for

Trading date time	Description
	transmission set out in Article 11 of RTS 2025/1140 were not satisfied, this shall be the date and time of the transaction rather than the time of the order transmission.
Required Field	Mandatory field at the level of validation rules which is part of the record's unique identifier: Trading platform for crypto-asset + Transaction Record Number (where applicable) + Trading platform for crypto-asset transaction identification code + Trading date time
Validation Rules	1. This field must not be null. 2. The same timestamp format must be used across the submitted files. i.e. if the adapter for the CASP is configured to parse "YYYY-MM-DD HH:mm:ss.fff", then submitting any record with "YYYY-MM-DD HH:mm:ss" would result in record rejection. 3. Trading date time will be checked for defaulted timestamps (e.g. midnight defaulted timestamps). 4. The system also identifies duplicate records based on the following criteria: a. Execution records are considered duplicates when more than one record has the same combination of Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time. b. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 4 - Trading date time

2. Transaction Record Number

Transaction Record Number	Description
Example	456321vvdgbj
RTS 2025/1140 Annex – Table 3 Field	[3.2]
RTS 2025/1140 Annex – Table 3 Field Name	Transaction Record Number
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/TxRcrdId
Type / Code	Text {1,52}
Field Description	Identification number that is unique to the executing firm for each record.

Transaction Record Number	Description
Required Field	Mandatory field at the level of validation rules which is part of the record's unique identifier: Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time
Validation Rules	1. This field must not be null. 2. Data format must conform to the Type/Code. 3. The system also identifies duplicate records based on the following criteria: a. Execution records are considered duplicates when more than one record has the same combination of Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time. b. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 5 - Transaction Record Number

3. Trading platform for crypto-asset transaction identification code

Trading platform for crypto-asset transaction identification code	Description
Example	1000003334455Bgf
RTS 2025/1140 Annex – Table 3 Field	[3.3]
RTS 2025/1140 Annex – Table 3 Field Name	Trading platform for crypto-asset transaction identification code
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/Tx/TradPlcMtchgld
Type / Code	text{1,52}
Field Description	A number generated by the trading platform for crypto-assets and disseminated to both the buying and the selling parties in accordance with Article 14 of Delegated Regulation (EU) 2025/416 supplementing Regulation (EU) 2023/1114. Where relevant, the transaction hash or other identification alphanumeric string which is automatically generated on the DLT that enables to uniquely identify a specific transaction.

Trading platform for crypto-asset transaction identification code	Description
Required Field	Conditionally mandatory field (mandatory only if MIC code is populated in the venue field) which is part of the record's unique identifier: Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time
Validation Rules	1. This field must not be null, when a MIC is populated 2. Data format must conform to the Type/Code. 3. The system identifies duplicate records based on the following criteria: a. Execution records are considered duplicates when more than one record has the same combination of Trading platform for crypto-asset (where applicable) + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time. b. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 6 - Trading platform for crypto-asset transaction identification code

4. Price currency

Price Currency	Description
Example	BTC/USDT
RTS 2025/1140 Annex – Table 3 Field	[3.32]
RTS 2025/1140 Annex – Table 3 Field Name	Price Currency
Logical Path	/Document/CrptAsstRcrdKeepgRpt /Tx/New/Tx/Pric/DgtlTknPric/DgtlTknQty/Idr/DTIShrtnm
Type / Code	text{1,50}
Field Description	Currency in which the price is expressed (applicable where the price is expressed as a monetary value). Where the price of the crypto-asset is expressed in monetary terms and is expressed in a currency pair, the currency pair in which the price for the crypto-asset related to the order is expressed shall be reported. The first currency code shall be that of the base currency and the second

Price Currency	Description
	currency code shall be that of the quote currency. The quote currency determines the price of one unit of the base currency. The ISO currency code and the DTI short name as registered in accordance with the ISO 24165-2 data elements for registration of the DTI or the alternative identifier referred to in Article 15 of RTS 2025/1140 shall be used to represent the fiat currency and the crypto asset respectively in the currency pair.
Required Field	Mandatory at the level of validation rule
Validation Rules	1. This field must not be null. 2. The value must be valid against ISO 24165-2 and the number format must conform to the Type/Code description.

Table 7 - Price Currency

5. Crypto-asset identification code

Crypto Asset Identification Code	Description
Example	K1NS41N51
RTS 2025/1140 Annex – Table 3 Field	[3.38]
RTS 2025/1140 Annex – Table 3 Field Name	Crypto Asset Identification Code
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/CrptAsst/DgtlTknIdr/DTI /Document/CrptAsstRcrdKeepgRpt/Tx/New/Tx/Qty/DgtlTknQty/Idr/DTI
Type / Code	text [1-9B-DF-HJ-NP-TV-XZ][0-9B-DF-HJ-NP-TV-XZ]{8,8}
Field Description	Digital token identifier as defined in ISO 24165 standard.
Required Field	Mandatory
Validation Rules	1. The field must not be null. 2. It must be valid against ISO 24165-2.

Table 8 - Crypto Asset Identification Code

6. Buyer / Seller identification code

Buyer / Seller identification code	Description
Example	BUYI
RTS 2025/1140 Annex – Table 3 Field	3.6 and 3.15
RTS 2025/1140 Annex – Table 3 Field Name	Buyer and/or Seller identification code
Logical Path	One or both elements below should be populated per record: Buy leg: Side value will be = 'BUYI' /Document/CrptAsstRcrdKeepgRpt/Tx/New/Buyr/AcctOwnr/Id/AltrnId/Id Sell leg: Side value will be = 'SELL' /Document/CrptAsstRcrdKeepgRpt/Tx/New/Sellr/AcctOwnr/Id/AltrnId/Id
Type / Code	String
Field Description	The side of the trade taken from the perspective of Client or Account
Required Field	Mandatory to report Buyer Leg or Seller Leg or both
Validation Rules	For trading venues, both matched trades must be submitted as one record. Both the required attributes nested in the "Buyer" and "Seller" objects must be provided in a single record. If the required attributes are not populated on either the buyer or seller side, one or both of the new trades will be rejected. Furthermore, across the submitted file, the trade count ratio between buys and sells must be within +- 0.1% proximity. Default value 'BUYI' will be used for element /Document/CrptAsstRcrdKeepgRpt/Tx/New/Buyr/AcctOwnr/Id/AltrnId/Id and default value 'SELL' will be used for element /Document/CrptAsstRcrdKeepgRpt/Tx/New/Sellr/AcctOwnr/Id/AltrnId/Id.

Table 9 - Side

7. Quantity

Quantity	Description
Example	5

Quantity	Description
RTS 2025/1140 Annex – Table 3 Field	[3.29]
RTS 2025/1140 Annex – Table 3 Field Name	Quantity
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New /Tx/Qty/DgtlTknQty/Unit
Type / Code	Unit
Field Description	The number of units of the crypto-assets. Where the price is expressed in sub-components of that crypto-asset, it shall be nonetheless recorded in decimal notation of the price expressed in units of that crypto-asset.
Required Field	Mandatory
Validation Rules	1. This field must not be null and it must be greater than zero. 2. This field must also conform to the Type/Code.

Table 10 - Quantity

8. Price

Price	Description
Example	90000.15
RTS 2025/1140 Annex – Table 3 Field	[3.31]
RTS 2025/1140 Annex – Table 3 Field Name	Price
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New /Tx/Pric/DgtlTknPric/DgtlTknQty/Unit
Type / Code	decimal td = 30 fd = 29 type: string pattern: <code>^[+-]?0*(((0[1-9][0-9]{0,29})([.]0*)?) ([.]0-9){1,29}0*) ((?=[1-9.]0-9.[1,30]0*\$)([1-9][0-9]*)?[.]0-9){0,29}0*)\$</code>
Field Description	Traded price of the transaction excluding, where applicable, commission, any other fee and accrued interest. Where the crypto-asset is traded based on a

Price	Description
	currency pair, the price shall express the quantity of the quote currency for one unit of the base currency. Where the price is expressed in sub-components of that crypto-asset, the price shall be nonetheless recorded in decimal notation of the price expressed in units of that crypto-asset. Where the price is recorded in monetary terms, the price shall be provided in the major currency unit. Where price is not applicable, the value shall be 'NOAP'.
Required Field	Mandatory
Validation Rules	1. This field must not be null and it must be greater than zero. 2. The field must also conform to the Type/Code. 3. The system checks for execution price discrepancy (e.g. 1,000% jump between two consecutive execution prices in the same Price Currency).

Table 11 - Price

9. Trading platform for crypto-asset

Trading platform for crypto-asset	Description
Example	KRME
RTS 2025/1140 Annex – Table 3 Field	[3.33]
RTS 2025/1140 Annex – Table 3 Field Name	Trading platform for crypto-asset
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/Tx/TradVn
Type / Code	text [A-Z0-9]{4,4}
Field Description	Identification of the trading platform for crypto-asset where the transaction was executed. The ISO 10383 segment MIC must be used for transactions executed on a trading platform for crypto-assets. Where the segment MIC does not exist, the operating MIC must be used. MIC code 'XOFF' should be used for crypto-assets admitted to trading

Trading platform for crypto-asset	Description
	or traded on a trading platform for crypto-assets or for which a request for admission was made, where the transaction on that crypto-asset was not executed on a trading platform for crypto-assets. MIC code 'XXXX' should be used for crypto-assets that are not admitted to trading or traded on a trading platform for crypto-assets, or for which no request for admission has been made. The system also identifies duplicate records based on the following criteria: Execution records are considered duplicates when more than one record has the same combination of Trading platform for crypto-asset (where applicable) + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.
Required Field	Mandatory field at the level of validation rules which is part of the record's unique identifier: Trading platform for crypto-asset + Transaction Record Number (where applicable) + Trading platform for crypto-asset transaction identification code + Trading date time
Validation Rules	1. This field value must be valid against the ISO 10383 list (except if the value is XOFF or XXXX). 2. This field must always be populated

Table 12 - Trading platform for crypto-asset

3.4.1.4 Description of orderbook business fields

Each orderbook business field is structured into standard attributes, as outlined in the table below:

List of Orderbook Fields Attributes	Attribute Description
Example	Execution field example
RTS 2025/416 Annex – Table 2 Field	Field number according to RTS 2025/416 Annex – Table 2
RTS 2025/416 Annex – Table 2 Field Name	Field name according to RTS 2025/416 Annex – Table 2
Logical Path	Location on JSON schema to report/distribute this element
Type / Code	Data type

List of Orderbook Fields Attributes	Attribute Description
Field Description	Description of the field
Required Field	Mandatory / Optional
Validation Rules	Specific validation rules to be applied to this field

Table 13 - Orderbook data attributes

3.4.1.5 List of orderbook fields

1. Date and time

Date and time	Description
Example	2026-03-15 14:25:36.125
RTS 2025/416 Annex – Table 2 Field	[2.8]
RTS 2025/416 Annex – Table 2 Field Name	Date and time
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt /New/Ordr/OrdrIdData/TmStmp
Type / Code	dateTime
Field Description	Date and time of each event/auction. Specifies the UTC timestamp (epoch time) when a trading instruction. For orders which are not new, Date and time should reflect the time of the status change, not the original order creation time. For Trade Surveillance, this time is expected to be within the last 24 hrs.
Required Field	Mandatory field which is part of the record unique identifier: Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable) + CASP Name (automatically created by the system)
Validation Rules	1. This field must not be null. 2. The same timestamp format must be used across the submitted files. i.e. if the adapter for the CASP is configured to parse "YYYY-MM-DD HH:mm:ss.fff", then submitting any record with "YYYY-MM-DD HH:mm:ss" would result in record rejection. 3. The system also identifies duplicate records based on the following criteria: a. OrderBook records are considered duplicates when more than one record has the same combination Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable).

Date and time	Description
	b. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 14 - Date and time

2. New order, order modification, order cancellation, order rejections, partial or full execution

Status	Description
Example	CAME
RTS 2025/416 Annex – Table 2 Field	[2.20]
RTS 2025/416 Annex – Table 2 Field Name	New order, order modification, order cancellation, order rejections, partial or full execution
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/Ord/OrdIdData/EvtTp/Cd or /Document/CrptAsstOrdRpt/OrdRpt/New/Ord/OrdIdData/EvtTp/Prtry/Id
Type / Code	Text (enumerated list) NEWO (New order) TRIG (Triggered) REME (Replaced by the member or participant of the trading platform for crypto-assets) REMA (Replaced by market operations (automatic)) REMH (Replaced by market operations (human intervention)) CHME (Change of status at the initiative of the participant of the trading platform for crypto-assets) CHMO (Change of status due to market operations) CAME (Cancelled at the initiative of the participant of the trading platform for crypto-assets) CAMO (Cancelled by market operations) REMO (Rejected order) EXPI (Expired order) PARF (Partially filled) FILL (Filled)

Status	Description
Field Description	New order: submission of a new order to the crypto-asset service provider operating the trading platform for crypto-assets. Triggered: an order which becomes executable or, as the case may be, non-executable upon the realisation of a pre-determined condition. Replaced by the participant of the trading platform for crypto-assets: where a participant or client of the trading platform for crypto-assets decides upon its own initiative to change any characteristic of the order it has previously entered into the order book. Replaced by market operations (automatic): where any characteristic of an order is changed by the trading platform for crypto-assets operator's ICT systems. This includes where a peg order's or a trailing stop order's current characteristics are changed to reflect how the order is located within the order book. Replaced by market operations (human intervention): where any characteristic of an order is changed by a trading platform for crypto-assets operator's staff. This includes the situation where a participant of the trading platform for crypto-assets requests to urgently cancel the orders linked to ICT incidents. Change of status at the initiative of the participant of the trading platform for crypto-assets. This includes activation and deactivation. Change of status due to market operations. Cancelled upon the initiative of the participant of the trading platform for crypto-assets: where a participant or client decides upon its own initiative to cancel the order it has previously entered. Cancelled by market operations. Rejected order: an order received but rejected by the operator of the trading platform for crypto-assets. Expired order: where the order is removed from the order book upon the end of its validity period. Partially filled: where the order is not fully executed so that there remains a quantity to be executed. Filled: where there is no more quantity to be executed.
Required Field	Mandatory
Validation Rules	1. This field must not be null and must be a part of the shortlist of acceptable 4-letter codes. 2. The system compares the mix of Status values in each submitted file against the historical patterns observed for the same crypto-asset service provider. Files whose composition deviates significantly from the entity's established reporting behaviour are flagged for review. This includes cases where a status that is normally present is missing from the submission, which may indicate that an event type is not being captured upstream.

Table 15 - New order, order modification, order cancellation, order rejections, partial or full execution

3. Order identification code

Order identification code	Description
Example	1000003334455Bgf
RTS 2025/416 Annex – Table 2 Field	[2.19]
RTS 2025/416 Annex – Table 2 Field Name	Order identification code
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/ New/Ordr/OrdrIdData/OrdrId
Type / Code	text{1,50}
Field Description	A unique identifier for the order. Alphanumerical code assigned by the operator of the trading platform for crypto-assets to the individual order.
Required Field	Mandatory field which is part of the record unique identifier: Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable) + CASP Name
Validation Rules	1. This field must not be null and the text length must be <= 50 chars. 2. The system also identifies duplicate records based on the following criteria: a. OrderBook records are considered duplicates when more than one record has the same combination of Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable). b. If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 16 - Order identification code

4. Transaction identification code

Transaction identification code	Description
Example	456321vvdgbj

Transaction identification code	Description
RTS 2025/416 Annex – Table 2 Field	[2.46]
RTS 2025/416 Annex – Table 2 Field Name	Transaction identification code
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/OrdRpt/OrdData/TxData/TxId
Type / Code	text{1,52}
Field Description	Alphanumeric code assigned by the trading platform for crypto-asset to the transaction pursuant to Article 14 of RTS 2025/416. The code shall be unique, consistent, and persistent per ISO10383 segment MIC and per trading day. The components of the transaction identification code shall not disclose the identity of the counterparties to the transaction for which the code is maintained. For orders transmitted to trading platforms for crypto-assets as referred to in Article 12 of Commission Delegated Regulation establishing technical standards adopted pursuant to Article 68(10), first subparagraph, point (b) to an entity providing crypto-asset services outside the Union, this information shall be recorded whenever those are retrievable.
Required Field	Conditionally Mandatory: Required when Status = "FILL" or "PARF" Field necessary to reconcile execution info with order info
Validation Rules	1. If "Status" is "PARF" or "FILL" this field must not be empty. Otherwise, this field must be empty. 2. The text length must be <= 52 chars.

Table 17 - Transaction identification code

5. Order type classification

Order type classification	Description
Example	LMTO
RTS 2025/416 Annex – Table 2 Field	[2.22]

Order type classification	Description
RTS 2025/416 Annex – Table 2 Field Name	Order type classification
Logical Path	/Document/CrptAsstOrdRBookRpt/OrdRpt/New /OrdR/OrdRData/OrdRClssfctn/OrdRTpClssfctn
Type / Code	text (enumerated) LMTO (LIMIT order) STOP (STOP order)
Field Description	Classification of the order according to two generic order types. 'LMTO' - LIMIT order: in the cases where the order is tradable, 'STOP' - STOP order: in the cases where the order becomes tradable only upon the realisation of a predetermined price event.
Required Field	Mandatory
Validation Rules	1. This field must not be null. 2. The value must be part of the shortlist of acceptable Order type classification values

Table 18 - Order type classification

6. Buy-sell indicator

Buy-sell indicator	Description
Example	BUYI
RTS 2025/416 Annex – Table 2 Field	[2.30]
RTS 2025/416 Annex – Table 2 Field Name	Buy-sell indicator
Logical Path	/Document/CrptAsstOrdRBookRpt/OrdRpt/New /OrdR/OrdRData/InstrData/BuySellInd
Type / Code	text (enumerated) BUYI (buy) SELL (sell)
Field Description	Indicates whether the order is to buy or sell. Possible values are: 'BUYI' - buy 'SELL' - sell
Required Field	Mandatory

Buy-sell indicator	Description
Validation Rules	1. This field must not be null. 2. "BUYI" or "SELL" are the only accepted values. Order count ratio between buys and sells should not be extreme (i.e. >90% are buy)

Table 19 - Buy-sell indicator

7. Initial quantity

Initial quantity	Description
Example	10.5
RTS 2025/416 Annex – Table 2 Field	[2.34]
RTS 2025/416 Annex – Table 2 Field Name	Initial quantity
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New /OrdR/OrdRData/InstrData/InitlQty/DgtlTknQty/Unit
Type / Code	decimal td = 30 fd = 29 type: string pattern: <code>^[+-]?0*(((0 [1-9][0-9]{0,29})([.]0*)?) ([.][0-9]{1,29}0*))(((?=[1-9.][0-9]{1,30}0*\$)([1-9][0-9]*)?[.][0-9]{0,29}0*))\$</code>
Field Description	The number of units of the crypto-asset in the order in case the quantity is expressed as a nominal value. It corresponds to quantity notation 'NOML' for field 25 of the records-keeping RTS. Or The number of units of the crypto-asset in the order expressed as a number of crypto-assets units. In case the order pertains a fraction of a crypto-asset, indicate the quantity in decimal notation of the unit. It corresponds to quantity notation 'CRYP' for field 25 of RTS 2025/416.
Required Field	Mandatory
Validation Rules	This field must not be null and it must be greater than zero. Furthermore, the number format must conform to the Type/Code.

Table 20 - Initial quantity

8. Price currency

Price currency	Description
Example	BTC/USDT
RTS 2025/416 Annex – Table 2 Field	[2.28]
RTS 2025/416 Annex – Table 2 Field Name	Price currency
Logical Path	/Document/CrptAsstOrdRBookRpt/OrdRpt/New/OrdR/OrdRData/OrdRPrics/LmtPric/DgtlTknPric/DgtlTknQty/Idr/DTIShrtNm
Type / Code	text{1,50}
Field Description	Identification of the currency pair in which the price related to the order is expressed (applicable where the price is expressed in monetary terms in a currency pair). Where the price of the crypto-asset is expressed in monetary terms and it is expressed in a currency pair, the currency pair in which the price for the crypto-asset related to the order is expressed shall be reported. The first currency code shall be that of the base currency and the second currency code shall be that of the quote currency. The quote currency determines the price of one unit of the base currency. The ISO currency code and the DTI short name as registered according to the ISO 24165-2 data elements for registration of the DTI or alternative identifier referred to in Article 15 shall be used to represent the fiat currency and the crypto-asset, respectively, in the currency pair. (DTI short name used to identity the currency pair)
Required Field	Mandatory
Validation Rules	This field must not be null and it must be valid against ISO 24165-2. Furthermore, the number format must conform to the Type/Code.

Table 21 - Price currency

9. Crypto-asset identification code

Crypto-asset identification code	Description
Example	K1NS41N51
RTS 2025/416 Annex – Table 2 Field	[2.17]
RTS 2025/416 Annex – Table 2 Field Name	Crypto-asset identification code
Logical Path	/Document/CrptAsstOrdRBookRpt/RptHdr/DgtlTknIdr/DTI

Crypto-asset identification code	Description
Type / Code	text [1-9B-DF-HJ-NP-TV-XZ][0-9B-DF-HJ-NP-TV-XZ]{8,8}
Field Description	Unique and unambiguous identifier of the base crypto-asset in accordance with Article 4.
Required Field	Mandatory
Validation Rules	This field must not be null and must be valid against ISO 24165-2.

Table 22 - Crypto-asset identification code

10. Remaining Quantity

Remaining quantity	Description
Example	8
RTS 2025/416 Annex – Table 2 Field	[2.35]
RTS 2025/416 Annex – Table 2 Field Name	Remaining quantity
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New /Ord/OrdData/InstrData/RmngQty /DgtITknQty/Unit
Type / Code	decimal td = 30 fd = 29 type: string pattern: <code>^[+-]?0*(((0 [1-9][0-9]{0,29})([.]0*)?) ([.][0-9]{1,29}0*))(((?=[1-9.][0-9.]{1,30}0*\$)([1-9][0-9]*)?[.][0-9]{0,29}0*))\$</code>
Field Description	The total quantity that remains in the order book after a partial execution or in the case of any other event affecting the order expressed in nominal value. On a partial fill order event, this shall be the total remaining volume after that partial execution. On an order entry this shall equal the initial quantity. It corresponds to quantity notation 'UNIT' for field 32 of RTS 2025/416.
Required Field	Conditionally Mandatory: Required when Status = 'FILL' or 'PARF'
Validation Rules	This field should not be null when Status = 'FILL' or 'PARF'. It must be greater than or equal to zero and lower or equal to "Initial

Remaining quantity	Description
	quantity". The field number format must conform to the Type/Code description.

Table 23 - Remaining quantity

11. Passive only indicator

Passive only indicator	Description
Example	0
RTS 2025/416 Annex – Table 2 Field	[2.41]
RTS 2025/416 Annex – Table 2 Field Name	Passive only indicator
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New /OrdRpt/OrdData/InstrData/PssvOnlyInd
Type / Code	boolean enum: ["1", "0", "true", "false"]
Field Description	Indicates if the order is submitted to the trading venue with a characteristic or flag, such that the order shall not immediately execute against any contra visible orders. Note: When absent, default value is false.
Required Field	Optional
Validation Rules	The number format must conform to the Type/Code description and it must be a part of the shortlist of acceptable values.

Table 24 - OrderPassiveOnly

12. Traded quantity

Traded quantity	Description
Example	2
RTS 2025/416 Annex – Table 2 Field	[2.37]
RTS 2025/416 Annex – Table 2 Field Name	Traded quantity
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New /OrdRpt/OrdData/TxData/TraddQty/DgtlTknQty/Unit
Type / Code	decimal

Traded quantity	Description
	td = 18 fd = 17 type: string pattern: <code>^[+-]?0*(((0 [1-9][0-9]{0,17})([.]0*)?) ([.][0-9]{1,17}0*))(((?=[1-9.][0-9.]{1,18}0*\$)([1-9][0-9]*)?[.][0-9]{0,17}0*))\$</code>
Field Description	Where there is a partial or full execution, this field shall be populated with the executed quantity expressed as a number of crypto assets. It corresponds to quantity notation 'CRYP' for field 32 of RTS 2025/416.
Required Field	Conditionally Mandatory: Required when Status = 'FILL' or "PARF"
Validation Rules	If "Transaction identification code" is filled, quantity must be greater than or equal to zero and lower or equal to "Initial quantity" If "Transaction identification code" is empty, this field must be empty. Furthermore, the number format must conform to the Type/Code.

Table 25 - Traded quantity

13. Limit price

Limit price	Description
Example	90000.15
RTS 2025/416 Annex – Table 2 Field	[2.23]
RTS 2025/416 Annex – Table 2 Field Name	Limit price
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New /Ordr/OrdrData/OrdrPrics/LmtPric/ DgtlTknPric/DgtlTknQty/Unit
Type / Code	"{DECIMAL-18/13} in case the price is expressed as monetary value. {DECIMAL-11/10} in case the price is expressed as a percentage or yield. {DECIMAL-18/17} in case the price is expressed as basis points. 'NOAP'"
Field Description	The maximum price at which a buy order can trade or the minimum price at which a sell order can trade.

Limit price	Description
	The spread price for a strategy order. It can be negative or positive. This field shall be 'NOAP' where not relevant. Where price is reported in monetary terms, it shall be provided in the major currency unit. If the crypto-asset is traded based on a currency pair the price shall express the quantity of the quote currency for one unit of the base currency. If the price is expressed in sub-components of that crypto-asset, it shall be nonetheless recorded in decimal notation of the price expressed in units of that crypto-asset.
Required Field	Mandatory:
Validation Rules	This field must not be null and must be greater than zero when order is a true limit order. Furthermore, the number format must conform to the Type/Code.

Table 26 - Limit price

14. Passive or aggressive indicator

Passive or aggressive indicator	Description
Example	PASV
RTS 2025/416 Annex – Table 2 Field	[2.42]
RTS 2025/416 Annex – Table 2 Field Name	Passive or aggressive indicator
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New OrdR/OrdRData/TxData/PssvOrAggrssvInd
Type / Code	text (enumerated) PASV (passive) AGRE (aggressive)
Field Description	Identifies whether the participant was passive or the aggressor. On partial fill and fill order events, indicates whether the order was already resting on the order book and providing liquidity (passive) or the order initiated the trade and thus took liquidity (aggressive). This field shall be left blank if not applicable. Possible values are: 'PASV' - passive

Passive or aggressive indicator	Description
	'AGRE' - aggressive
Required Field	Optional
Validation Rules	This field should only be populated with one of the authorised values if order "Status" is "PARF" or "FILL" and must be empty otherwise.

Table 27 - Passive or aggressive indicator

15. Validity period

Validity period	Description
Example	GTCV
RTS 2025/416 Annex – Table 2 Field	[2.9]
RTS 2025/416 Annex – Table 2 Field Name	Validity period
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New /Ordr/OrdrIdData/VldtyPrd/VldtyPrdCd
Type / Code	text (enumerated) DAVY (Good-For-Day) GTCV (Good-Till-Cancelled) GTTV (Good-Till-Time) GTDV (Good-Till-Date) GTSV (Good-Till-Specified Date and Time) GATV (Good After Time) GADV (Good After Date) GASV (Good After Specified Date and Time) IOCV (Immediate-Or-Cancel) FOKV (Fill-Or-Kill)
Field Description	Code defining the validity period of the order (Good-for-day, Good-till-time/date, etc.). Possible values are: 'DAVY' - Good-For-Day 'GTCV' - Good-Till-Cancelled 'GTTV' - Good-Till-Time 'GTDV' - Good-Till-Date

Validity period	Description
	'GTSV' - Good-Till-Specified Date and Time 'GATV' - Good After Time 'GADV' - Good After Date 'GASV' - Good After Specified Date and Time 'IOCV' - Immediate-Or-Cancel 'FOKV' - Fill-Or-Kill
Required Field	Mandatory
Validation Rules	This field must not be null and the value must be a part of the shortlist of acceptable 4-letter codes.

Table 28 - Validity period

16. Validity period and time

Validity period and time	Description
Example	2026-03-15 17:25:36.125
RTS 2025/416 Annex – Table 2 Field	[2.11]
RTS 2025/416 Annex – Table 2 Field Name	Validity period and time
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New /Ordr/OrdrIdData/VldtyDtTm
Type / Code	dateTime
Field Description	Timestamp reflecting the time on which the order becomes active or it is ultimately removed from the order book. Good for day: the date of entry with the timestamp immediately prior to midnight. Good till time: the date of entry and the time to that specified in the order. Good till date: will be the specified date of expiry with the timestamp immediately prior to midnight. Good till specified date and time: the specified date and time of expiry. Good after time: the date of entry and the specified time at which the order becomes active. Good after date: the specified date with the timestamp immediately after midnight.

Validity period and time	Description
	Good after specified date and time: the specified date and time at which the order becomes active. Good till Cancel: the ultimate date and time the order is automatically removed by market operations. Other: timestamp for any additional validity type.
Required Field	Conditionally Mandatory: When 'Validity period' is "GTDV" then Validity period and time is mandatory.
Validation Rules	This field must not be null if 'Validity period' is "GTDV". The same timestamp format must be used across the files. i.e. if the adapter for the CASP is configured to parse "YYYY-MM-DD HH:mm:ss.fff", then any record with "YYYY-MM-DD HH:mm:ss" will be rejected.

Table 29 - Validity period and time

17. Segment MIC

Segment MIC	Description
Example	KRME
RTS 2025/416 Annex – Table 2 Field	[2.15]
RTS 2025/416 Annex – Table 2 Field Name	Segment MIC
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New /Ordr/OrdrldData/TradVn
Type / Code	text [A-Z0-9]{4,4}
Field Description	Identification of the trading platform for crypto-asset where the order was submitted. If the trading platform for crypto-asset uses Segment MICs then the Segment MIC shall be used. If the trading platform for crypto-asset does not use segment MICs then the operating MIC shall be used.
Required Field	Optional field which is part of the record unique identifier: Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable) + CASP Name
Validation Rules	If "Transaction identification code" is filled, this field should be valid against ISO 10383 list (except if XOFF or XXXX).

Segment MIC	Description
	If "Transaction identification code" is empty, then this field must be empty. The system also identifies duplicate records based on the following criteria: OrderBook records are considered duplicates when more than one record has the same combination of Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable). If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.

Table 30 - Segment MIC

18. Transaction price

Transaction price	Description
Example	90000.15
RTS 2025/416 Annex – Table 2 Field	[2.27]
RTS 2025/416 Annex – Table 2 Field Name	Transaction price
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New Ordr/OrdrData/TxData/TxPric/ DgtlTknPric/DgtlTknQty/Unit
Type / Code	decimal td = 30 fd = 29 type: string pattern: ^[+-]?0*(((0 [1-9][0-9]{0,29})([.]0*)?) ([.][0-9]{1,29}0*))((?=[1-9.][0-9.]{1,30}0*\$)([1-9][0-9]*)?[.][0-9]{0,29}0*)\$
Field Description	Traded price of the transaction excluding, where applicable, commission, other fees and accrued interest in case the price is expressed in number of tokens or the crypto-asset is traded based on a currency pair. If the crypto-asset is traded based on a currency pair the price shall express the quantity of the quote currency for one unit of the base currency. If Status = Filled, then the execution price of the fill.

Transaction price	Description
Required Field	Conditionally Mandatory: Required when Status = 'FILL' or 'PARF' and the order is a true limit order.
Validation Rules	This field must not be null when Status = 'FILL' or 'PARF' and the order is a true limit order. This must be greater than zero and within the range defined by "Limit price" if Limit price field is not empty. If "Transaction identification code" is empty, this field must be empty.

Table 31 - Transaction price

3.4.2 Additional Mandatory fields at JSON syntax level

Apart from the business fields defined in 3.4.1 Business fields, there are additional fields that must be reported for Execution and OrderBook submissions. While these fields are not processed by the MIDAS system for alert generation purposes, they are required by the ESMA MiCA order book and records keeping message specification and the JSON schema provided in section 0

Annex 2: File format: JSON schema. Since all submitted files undergo schema validation to confirm compliance with the published schema, any submission that omits these fields will fail schema validation and be rejected. The fields listed in this section are currently not subject to content validation rules within MIDAS. As the system evolves, dedicated validation rules covering these fields will be introduced in future releases.

3.4.2.1 Execution additional fields

1. Seller Branch Indicator

Branch Indicator	Description
JSON element	BrnchInd
RTS 2025/1140 Annex – Table 3 Field	[3.16]
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/Sellr/AcctOwnr/BrnchInd
Field Description	Where the seller is a client, indicates whether a branch received the order from a client or made an investment decision for a client in accordance with a discretionary mandate given to it by the client as required by Article 16 of RTS 2025/1140. Possible values are: 'TRUE' - True 'FALS' - False 'NOAP' - Not applicable Note: NOAP should be reported for the Seller Leg
Required Field	Conditional Element Mandatory when Seller identification is reported
Validation Rules	

Table 32 - Seller Branch Indicator

2. Buyer Branch Indicator

Branch Indicator	Description
JSON element	BrnchInd
RTS 2025/1140 Annex – Table 3 Field	[3.7]
RTS 2025/417 Annex II – Table 2 Field	N/A
Logical Path	/Document/CrptAsstRcrdKeepgRpt/Tx/New/Buyr/AcctOwnr/BrnchInd

Branch Indicator	Description
Field Description	Where the buyer is a client, indicates whether a branch received the order from a client or made an investment decision for a client in accordance with a discretionary mandate given to it by the client as required by Article 16 of RTS 2025/1140. Possible values are: 'TRUE' - True 'FALS' - False 'NOAP' - Not applicable Note: NOAP should be reported for the Buyer Leg
Required Field	Conditional Element Mandatory when Buyer identification is reported
Validation Rules	

Table 33 - Buyer Branch Indicator

3.4.2 Orderbook additional fields

1. Report Header Reporting Entity LEI

LEI	Description
JSON element	LEI
RTS 2025/416 Annex – Table 2 Field	N/A
Logical Path	/Document/CrptAsstOrdrBookRpt/RptHdr/RptgNtty/LEI
Type / Code	text [A-Z0-9]{18,18}[0-9]{2,2}
Field Description	Legal entity identifier of the party. Legal Entity Identifier of the CASP which generates the report.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 34 - Report Header Reporting Entity LEI

2. Report Header Date

Date	Description
JSON element	Dt

Date	Description
RTS 2025/416 Annex – Table 2 Field	N/A
Logical Path	/Document/CrptAsstOrdrBookRpt/RptHdr/RptgPrd/Dt
Type / Code	date
Field Description	Date period is limited to a single date.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 35 - Report Header Date

3. Report Identification

Report Identification	Description
JSON element	RptId
RTS 2025/416 Annex – Table 2 Field	N/A
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New/RptId
Type / Code	text{1,140}
Field Description	Unique and unambiguous technical identification of the order report. Usage: Used for reconciliation purpose in case of cancellation.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 36 - Report Identification

4. Order Book Identification

Order Book Identification	Description
JSON element	OrdrBookId
RTS 2025/416 Annex – Table 2 Field	[2.16]
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New/Ordr/OrdrIdData/OrdrBookId
Type / Code	Text {1,35}

Order Identification	Book	Description
Field Description		Alphanumeric code established by the trading platform for crypto-assets for each order book. Not used for purpose of market abuse generation.
Required Field		Mandatory Element as per JSON syntax
Validation Rules		

Table 37 - Order Book Identification

5. Sequence Number

Sequence Number	Description
JSON element	SeqNb
RTS 2025/416 Annex – Table 2 Field	[2.14]
Logical Path	/Document/CrptAsstOrdRBookRpt/OrdRpt/New/OrdR/OrdRldData/SeqNb
Type / Code	0 <= decimal td = 50 fd = 0 type: string pattern: ^[+]?0*(((0 [1-9][0-9]{0,49})([.]0*)?)[.][0]+)\$
Field Description	Each event is identified using unique positive integers in ascending order. The sequence number shall be unique to each type of event, consistent across all events, timestamped by the operator of the trading platform for crypto-assets, persistent for the date that the event occurs. Not used for purpose of market abuse generation.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 38 - Sequence number

6. Client Exception Identification

Exception Identification	Description
JSON element	XcptnId

Exception Identification	Description
RTS 2025/416 Annex – Table 2 Field	2.3
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/OrdRptData/CInt/XcptnId
Type / Code	text (enumerated) NOAP (Not applicable) PNAL (Pending allocation)
Field Description	Reason for not being able to identify the client of the participant to the trading platform for crypto-assets. This field shall be Not applicable ('NOAP') as it is not relevant.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 39 - Client Exception Identification

7. Non Executing Broker Party Exception

Party Exception	Description
JSON element	PtyXcptn
RTS 2025/416 Annex – Table 2 Field	[2.6]
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/OrdRptData/NonExctgBrkr/PtyXcptn
Type / Code	text (enumerated) NOAP (Not applicable)
Field Description	The code used to identify a participant of the trading platform for crypto-assets that routed an order on behalf of and in the name of another participant of the trading platform for crypto-assets. This field shall be Not applicable ('NOAP') as it is not relevant.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 40 - Non Executing Broker Party Exception

8. Trading Capacity

Trading Capacity	Description
JSON element	TradgCpcty
RTS 2025/416 Annex – Table 2 Field	[2.7]
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/OrdR/OrdRData/TradgCpcty
Type / Code	text (enumerated) DEAL (Exchanging crypto-assets for funds) MTCH (Matched principal) AOTC (Any other capacity)
Field Description	Indicates whether the crypto-asset service provider undertaking the transaction is carrying out matched principal trading or exchanging crypto-assets for funds. Where the order submission does not result from the crypto-asset service provider carrying out matched principal trading or exchanging crypto-assets for funds, the field shall indicate that the transaction was carried out under any other capacity. Possible values are: 'DEAL' - Exchanging crypto-assets for funds 'MTCH' - Matched principal 'AOTC' - Any other capacity
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 41 - Trading Capacity

9. Stop Price (No Price - Pending)

Stop Price Pending	Description
JSON element	Pdg
RTS 2025/416 Annex – Table 2 Field	2.25
Logical Path	/Document/CrptAsstOrdRpt/OrdRpt/New/OrdR/OrdRData/OrdRPrics/StopPric/NoPric/Pdg
Type / Code	text (enumerated) NOAP (Not applicable)
Field Description	The price that must be reached for the order to become active. This field shall be Not applicable ('NOAP') as it is not relevant.

Stop Price Pending	Description
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 42 - Stop Price (No Price - Pending)

10. Additional Limit Price (No Price – Pending)

Additional Limit Price Pending	Description
JSON element	Pdg
RTS 2025/416 Annex – Table 2 Field	2.24
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New/Ordr/OrdrData/OrdrPrics/AddtlLmtPric/NoPric/Pdg
Type / Code	text (enumerated) NOAP (Not applicable)
Field Description	Any other limit price which may apply to the order. This field shall be Not applicable ('NOAP') as it is not relevant.
Required Field	Mandatory Element as per JSON syntax
Validation Rules	

Table 43 - Additional Limit Price (No Price – Pending)

11. Pegged Price (No Price - Pending)

Pegged Price Pending	Description
JSON element	Pdg
RTS 2025/416 Annex – Table 2 Field	2.26
Logical Path	/Document/CrptAsstOrdrBookRpt/OrdrRpt/New/Ordr/OrdrData/OrdrPrics/PggdPric/NoPric/Pdg
Type / Code	text (enumerated) NOAP (Not applicable)
Field Description	The maximum price at which a pegged order to buy can trade or the minimum price at which a pegged order to sell can trade. This field shall be Not applicable ('NOAP') as it is not relevant.
Required Field	Mandatory Element as per JSON syntax

Pegged Price Pending	Description
Validation Rules	

Table 44 - Pegged Price (No Price - Pending)

3.5 Data validation

MIDAS applies a structured, multi-layered validation process to ensure that all incoming data is complete, accurate, consistent, and aligned with predefined schemas before it is made available for downstream processing, analytics, and reporting. This process is designed to proactively identify data quality issues early, reduce operational risk, and provide clear visibility into any errors or anomalies detected during ingestion.

3.5.1 Data transmission/file validation rules

Once a file is ingested in the system, MIDAS performs a series of structural checks to confirm that the file can be reliably interpreted and processed. These initial checks determine whether an issue triggers a total file-level rejection.

3.5.1.1 JSON file Validation

- **File Naming Convention:** MIDAS reviews the JSON file name to ensure alignment with the following naming convention: `<FileType>_<DateDDMMYYYY>_<partN>_<partM>.json`. Submitted files failing to match this naming convention are immediately rejected.
- **JSON Schema:** The JSON file is validated against the published JSON schema provided in

- **Annex 2: File format: JSON schema.** If a file is structurally malformed, corrupted, breaks core JSON syntax rules, or violates the JSON schema, then the entire file will be rejected, and no records within will be processed.

Only the fields used for the generation of alert mechanisms will be subsequently ingested and validated by MIDAS.

3.5.1.2 Required Fields Presence

MIDAS verifies that all fields required for the generation of MIDAS alert mechanisms, as defined in the JSON schema, are structurally present for each record block within the submitted JSON file.

3.5.2 Data content validation rules

For each individual record in the submitted JSON file that passed the validation checks described in section 3.5.1 Data transmission/file validation rules, MIDAS validates the content of every field to ensure data accuracy at the field level. The system validates each field against data types, expected logical ranges, internal dependencies, and external standard registries. If a content validation rule is triggered for a record, then that specific record will be rejected.

3.5.2.1 Field Format Validation

- **Mandatory fields:** The system verifies that all mandatory field values required for alert production are present in each record. Mandatory execution and orderbook fields marked as required cannot be null. If a missing value is submitted for a required field, a validation error is triggered, and the individual record will be rejected.
- **Data type:** Each field value is checked to ensure it matches its defined data type. For numerical price and quantity fields, pattern checks are executed to ensure alignment with maximum decimal digit criteria, preventing records with malformed numerical inputs from entering the system.
- **String length constraints:** Text type fields must strictly obey defined length thresholds. Any record containing values exceeding these structural boundaries will be automatically rejected or flagged as a data quality warning.
- **Pre-defined set of values** are verified directly at the JSON schema level. This structural schema check ensures that these fields (e.g. Buy-sell indicator, Validity period, Order type classification, Passive or aggressive indicator) contain only valid pre-defined values. Field values deviating from these enumerations will break structural schema conformance and trigger a record rejection.

3.5.2.2 Date and Time Validation

- Date and timestamp fields are validated to ensure temporal accuracy and consistency:
 - Correctness of the format.
 - Values within acceptable date ranges as defined by the system.
- All date and time values are expected to be submitted with millisecond precision. If the ingestion adapter for a CASP is configured to parse a specific sub-second mask (e.g. YYYY-MM-DD HH:mm:ss.fff), all the submitted records that drop this precise formatting result in an automatic record rejection.
- In addition, the following temporal dependency rules apply:
 - When Validity period is set to GTDV, the Validity period and time field is mandatory and cannot be null or empty. Records where this condition is not met will be rejected.
 - Files where a significant proportion of Trading date time values are set to midnight (00:00:00.000) are flagged as a data quality warning, as this pattern is indicative of a timestamp defaulting issue in the upstream system.

3.5.2.3 Value Constraints

Numeric fields (e.g. Quantity, Price, and Initial quantity) that are expected to be non-negative are validated to ensure that values are positive, helping to catch logical or calculation-related data issues. In case a negative value is detected in fields specifying absolute execution quantities, order volumes, or market prices, the individual record is rejected. For Remaining quantity and Traded quantity fields, a value of zero is permitted but they must not contain negative values.

- Sequential and Macro-Ratio Protections: The system performs statistical and time-series monitoring during data load.
 - The system compares the mix of Status values in each submitted file against the historical patterns observed for the same crypto-asset service provider. Files whose composition deviates significantly from the entity's established reporting behaviour are flagged for review. This includes cases where a status that is normally present is missing from the submission, which may indicate that an event type is not being captured upstream.
 - Price Jump Protection: The system runs chronological sequence checks on execution prices for the same asset; records demonstrating a price variance jump of 1,000% or greater between consecutive entries under the same Symbol are automatically flagged for review.

- Fields tied to established international operational consensus standards are validated against active official global registers:
 - Market Identification Codes (MIC): Segment MIC is verified against the official ISO 10383 directory, requiring a valid segment MIC (or an operating MIC if a distinct segment code is unavailable). Explicit regulatory exceptions are allowed for XOFF and XXXX. For orderbook records this check applies only when Transaction identification code field is populated. For execution records it applies unconditionally.
 - Currency Classifications: Fiat currency representations must resolve exactly against the active ISO 4217 standard currency codes.
 - Digital Token Identifiers (DTI): Digital asset tracking fields must map directly to the official ISO 24165-2. Records failing this check are flagged as a data quality warning.

3.5.2.4 Field Dependency Validation

The system enforces cross-field relational dependencies based on transaction statuses. This includes verifying that:

- Transaction identification code is populated when Status is FILL or PARF and remains completely empty for all other statuses.
- Traded quantity, Transaction price, Trading platform for crypto-asset are populated if and only if Transaction identification code is populated.
- Traded quantity and Remaining quantity do not mathematically exceed the Initial quantity.
- When Status is FILL or PARF, fields like Remaining quantity are not null. Furthermore, if the order is a true limit order (Order type classification = LMTO), Limit price and Transaction price must be populated, with Transaction price falling strictly within the boundaries established by the Limit price.
- The Passive or Aggressive Indicator field must remain completely empty unless the Status is FILL or PARF.
- For trading venues, both matched trades must be submitted as one record, including both the Buy leg (with the Buyer alternative identification element populated) and the Sell leg (with the Seller alternative identification element populated).

The above will result in a record rejection or warning depending on the case (see Annex 3: Data validations for the full details).

3.5.2.5 Duplicate Detection

MIDAS identifies duplicate records to prevent double-counting or inconsistent reporting. If a file contains duplicate records, only the unique records will be processed. The duplicates will be ignored, but the file itself will still be processed for valid records. The trailing duplicate records will be rejected.

3.5.3 Feedback files

After processing each uploaded file, a feedback file is generated and written to the response directory of the CASP's SFTP account, typically within 2 hours of receipt. CASPs retrieve it using the same SFTP connection used to submit data. This file contains:

- **Transmission/file validation feedback:** In case a file fails the transmission/file validation rules defined in section 7.1 Transmission/file Validation Rules then the file will be rejected and no records will be processed. CASPs must correct the file and resubmit it in full.
- **Record-level (content) validation feedback:** In case specific records fail the content checks defined in section 7.2 Content Validation Rules, the feedback identifies the record and the content validation rule violated. For submissions containing multiple records, content validation is performed on a record level. Therefore, some records may be accepted while others may be rejected. The feedback provided is specific to each individual record and identifies the validation rule it violated.
- **Accepted record feedback:** For successful submissions, the feedback will indicate the acceptance of the records that triggered no validation rules.

The feedback file is delivered to the SFTP location for each CASP and remains there for the same period as the records and orders (7 years). Once the processing of a submission is finalized, the feedback file is written to the [response] directory of the submitting CASP's SFTP account, typically within 2 hours of receipt. CASPs retrieve it by connecting to their SFTP account and downloading the file from this directory — the same connection used to submit files. No email is sent, and no separate notification is generated; it is the CASPs' responsibility to check the response directory. An automated polling job is recommended on a fixed interval rather than manual checks. Details on the feedback file, along with a sample, are found in Annex 5.

3.5.4 Error correction process

1. Review the feedback file promptly after every submission.
2. Identify each rejected record and the rule it violated (record-level rejections) or the reason the file was rejected (file-level rejection).
3. Correct the issues at source.
4. Resubmit an updated version of the file that addresses all validation errors identified. For record-level rejections, resubmit only the corrected records; for a file-level rejection, resubmit the corrected file in full. Full list of error codes is included in Annex 3.

3.5.4.1 Common issues and how to resolve them

- **Malformed file structure.** Structural defects such as stray delimiters (e.g. extra commas) can cause records to error out. Validate file structure against the schema before submitting.
- **Missing mandatory fields.** Records missing any required field will be rejected; confirm all mandatory fields are populated.

3.5.4.2 Corrections and amendments to previously accepted records

Correcting a previously accepted record (e.g. a fat-finger error).

Submit the correction as a new record in your next data feed and provide ESMA /MIDAS support (support@esma.europa.eu) **advance notice** of the records to be overridden by emailing support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:Operations xxxx] on the email subject header. The prior version is retained in the system for audit purposes.

3.5.4.3 Duplicate records and resubmissions

- Duplicate checks are performed against the entire MIDAS historical database, not only within the submitted file. Duplicates are identified based on the following:

Orders: Unique key:

Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable) + CASP Name (automatically entered by the System)

For replaced orders with the same exact TransactTime (not date), the following fields are also considered: Initial quantity and Limit price.

Executions:

Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time

- If you need to resubmit a file unchanged following an operational issue, please notify MIDAS support by emailing support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:Operations xxxx] on the email subject header, with at least 3 business days' notice being sure to denote specific details of which file(s) may require being re-submitted. Otherwise, the records will be treated as duplicates and dropped.

3.5.4.4 Key reminders

Check your SFTP response directory after every submission (ideally via automated polling).

Correct all flagged errors before resubmitting.

Notify ESMA /MIDAS support with clear information by emailing support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:Operations xxxx] on the email subject header and with the appropriate lead time before resubmitting an unchanged file or overriding a previously accepted record.

4 Onboarding and Upload specifications

4.1 Onboarding Process

CASPs can request to commence the onboarding process by emailing MIDAS support at support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:New Onboarding Request] on the email subject header and providing the email addresses of the CASP's team members who will participate in the onboarding process, as well as IP addresses and public SSH keys.

The onboarding is overseen by ESMA and the relevant NCAs and is performed by the MIDAS system provider. The onboarding process commences with a meeting with ESMA and the MIDAS system provider. During this meeting the teams from the CASP, ESMA and the MIDAS system provider will set up a separate technical setup communication channel for the duration of this onboarding project.

The key stages are:

- Establish Access, Credentials, dedicated SFTP endpoint set-up
- Prepare JSON file
- UAT
- Release to Production

4.1.1 Establish Access

CASPs upload trade data JSON files to MIDAS via a secure SFTP drop point. Files land in a private AWS S3 bucket and are retrieved automatically for ingestion into MIDAS.

Initial Set-up:

1. Along with the initial onboarding request, CASPs will provide the IP address(es) that will be used by their organization to upload files, along with the CASP's public SSH keys. CASPs will send this information to MIDAS support at support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:New Onboarding Request]. The MIDAS Provider will whitelist the IP address(es) in the MIDAS system.
2. For each CASP, the MIDAS Provider will provision two dedicated SFTP endpoints—one for UAT and one for Prod—in the following format: `caspname@sftp<midas domain uat>` and `caspname@sftp<midas domain prod>`. The MIDAS Provider will communicate the SFTP endpoints and confirm readiness directly to the CASP.
3. From the approved and whitelisted IP addresses, CASPs will upload the execution (and orders if applicable) JSON file(s), (see section 4.1.2) to the SFTP UAT endpoint. It is suggested that a small representative sample of the file (a small data set representing common trades and

asset types) be uploaded, and the MIDAS Provider team will inform the CASP of any errors or potential issues, before commencing regular uploads. Full details and credentials to be further provided during a meeting with the MIDAS system provider.

4.1.2 Prepare JSON file

CASPs to prepare a representative sample of the file (a small data set representing the common trades and asset types)

- **File format:** Files should be in plain text JSON.
- **File size limit:** The MIDAS SFTP system accepts files up to 500 MB each. For larger data, the file must be split into parts to stay under the limit. When sending multiple files, processing will be more efficient if each file is roughly the same size (ideally 50–300 MB), rather than sending many small files. Files specifications are detailed in section 3.2.

CASPs must supply all fields marked as mandatory. Key points:

- **Record structure:** Each JSON file is an object containing data records. The exact structure is defined by the JSON schema for that file type.
- **Field requirements:** Each MIDAS field name has a mandatory indicator, format, and valid values. CASPs must ensure:
 - Mandatory fields are present and non-empty.
 - Data types match (e.g. numeric fields are numbers, timestamps in epoch ms or ISO datetime as specified).
 - Code lists are adhered to (e.g. Buyer / Seller identification code only "BUYI" or "SELL")
- **Record unique identification:** The combination of elements for a submitted record to be uniquely identified are:

Execution records:

1. Trading platform for crypto-asset
2. Trading date time
3. Trading platform for crypto-asset transaction identification code
4. Transaction Record Number

OrderBook records:

1. Segment MIC (if applicable)
2. Date and time
3. Order identification code
4. Status
5. Transaction identification code (if applicable)

CASPs should verify their data against the schema before submission. The MIDAS system will reject records that violate field formats or required codes. See Annex 1 for the list of fields and their types. See Annex 3 for content validation rules.

4.1.3 Upload of test sample JSON file(s) on UAT

CASPs to commence testing with uploading a representative sample of the file (a small data set representing the common trades and asset types). MIDAS system provider along with CASP performs Validation rules check and error handling.

- **Error handling:** The MIDAS SFTP server provides feedback if a file transfer fails (e.g. authentication error, disk quota exceeded). After transfer, if a file fails transmission or format checks, MIDAS will return an error code in the feedback file. In that case, the issue must be fixed and the corrected file must be re-uploaded. Details on the feedback file, along with a sample are found in Annex 5.
- After the initial testing with a representative sample file is completed, testing with full data set will follow in UAT.

4.1.4 Release to Production

After testing in UAT is successful, the MIDAS Provider will receive approval from CASP's relevant NCA to move the file feed to production. MIDAS Provider will provision a dedicated SFTP endpoint for Production in the following format: `caspname@sftp<midas domain prod>` and will communicate this information to CASP directly.

CASPs to commence uploading on a set date, while the MIDAS system provider, along with the CASP, performs the validation rule checks and error handling.

4.2 Communications

CASPs can address any questions related to this Reporting Instructions document process by emailing MIDAS support at support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:Reporting Instructions] on the email subject header.

4.3 Support

Once the CASP feed is released to Production, for any further operational support (for example for corrections and amendments to a previously accepted file) CASPs will receive support by emailing support@esma.europa.eu, adding [MIDAS][CASP][CASPName][Topic:Operations xxxx] on the email subject header.

5 Annex 1: Field type definitions

The following types are used within the MIDAS JSON Schema:

Symbol	Definition
{ALPHANUM-n}	Alphanumeric string up to n characters.
{DECIMAL-n/m}	Decimal with n total digits, m fraction digits (period "." as separator).
{DATE_TIME_FORMAT}	ISO 8601 format: YYYY-MM-DDThh:mm:ss.dddddd (UTC) or YYYY-MM-DD HH:mm:ss.fff
{ISIN}	12-character identifier (ISO 6166).
{LEI}	20-character identifier (ISO 17442).
{MIC}	4-character market identifier (ISO 10383).
{DTI}	Digital Token Identifier (ISO 24165)
{CCY}	Currency Code (ISO 4217)
{ENUM_4}	Strict 4-letter operational shortlist status codes
{BOOLEAN}	Logical state indicator accepting values

Table 45 - Field type definitions

6 Annex 2: File format: JSON schema



CryptoAssetRecordK
eeepingReport-auth11'



CryptoAssetOrderBo
okReport-auth118-V1.

7 Annex 3: Data validations

7.1 Transmission/file Validation Rules

Error code	Error Message	Control
FIL-101	Invalid file naming convention	The system ensures the JSON file name follows the mandatory format: <FileType>_<DateDDMMYYYY>_<partN>_<partM>.json. If the filename does not match this convention, the file is rejected before further processing.
FIL-102	JSON Schema validation failure	The JSON file is validated against the MIDAS JSON schema defined in Annex 2. The system checks the structural integrity (nesting, hierarchy, and syntax) of the file. Any structural deviation triggers this error.

Table 46 - Transmission/file validation rules

7.2 Content Validation Rules

7.2.1 Rejection validation rules

Error code	Error Message	Validation Outcome	Control
MET-001	Missing required field: [Field Name]	Rejection	The system verifies that all mandatory fields, as defined in the JSON schema defined in section 0 Annex 2: File format: JSON schema, are not null in the record. If a required field is null, this validation error is triggered for that specific record.
MET-002	Invalid field format or data type for [Field Name]	Rejection	Each field is validated to ensure it matches the expected data type and format (e.g., numeric, string, date, timestamp, etc.). This prevents malformed or incompatible data from entering the system.
MET-003	Value out of range for [Field Name].	Rejection	Fields are validated to ensure that their values fall within the acceptable ranges defined by the MIDAS system specifications.

MET-004	Negative or zero value detected in strictly positive field: [Field Name]	Rejection	For numeric fields expected to be strictly greater than zero (e.g., Quantity, Price, Initial quantity) the system performs a range check. If a negative value or a value of zero is detected, the record is rejected.
MET-005	Duplicate records detected in the submitted file	Rejection (The first record will be accepted and the rest of the records will be rejected)	The system identifies duplicate records. The system identifies duplicate records based on the following criteria: Execution records are considered duplicates when more than one record has the same combination of Trading platform for crypto-asset + Transaction Record Number + Trading platform for crypto-asset transaction identification code + Trading date time. OrderBook records are considered duplicates when more than one record has the same combination of Order identification code + Date and time + Status + Transaction identification code (if applicable) + Segment MIC (if applicable) + CASP Name If a file contains duplicate records, only the first record will be accepted while subsequent identical records will be rejected.
MET-006	Timestamp schema mismatch for [Field name]. Record timestamp format does not conform to the configured schema or differs from other record date/time values in the submitted file.	Rejection	The ingestion adapter for each CASP expects a single, strictly configured timestamp format across the entire file for Trading date time and Validity period and time values (e.g., YYYY-MM-DD HH:mm:ss.fff). Any record deviating from this configured format, including by dropping sub-second precision (e.g., YYYY-MM-DD HH:mm:ss), the record is structurally rejected.
MET-007	Validity period and time must not be null or empty if Validity period is GTDV.	Rejection	When 'Validity period' is "GTDV" then Validity period and time is mandatory.
MET-008	Validity period and time must be greater than Date and time.	Rejection	Validity period and time must represent a timestamp chronologically after the Date and time field value.

MET-009	Transaction identification code cannot be empty when Status is FILL or PARF.	Rejection	The system verifies that if the Status field is populated as PARF or FILL, the Transaction identification code, Remaining quantity and Traded quantity fields must not be null or empty in the record.
MET-010	Transaction identification code must be empty if the value of Status is different than FILL or PARF.	Rejection	The system ensures that for all execution statuses other than PARF or FILL, the Transaction identification code field must be completely blank/empty.
MET-011	[Segment MIC / Trading platform for crypto-asset must be empty when Transaction identification code is not populated.	Rejection for Traded quantity and Transaction price	If the Transaction identification code field is empty, Traded quantity, Transaction price fields must also be blank/empty. This rule only applies to Orderbook records.
MET-012	Invalid Range: Traded quantity exceeds logical boundaries.	Rejection	If Transaction identification code is populated, the system verifies that Traded quantity is greater than or equal to zero and less than or equal to Initial quantity
MET-013	Missing Required Field: Limit price / Transaction price	Rejection	The system verifies that when Status is equal to FILL or PARF and the order is a true limit order (Order type classification = LMTO), both Limit price and Transaction price fields cannot be null or empty.
MET-014	Mathematical Bounds Violation: Remaining quantity exceeds Initial quantity.	Rejection	The system enforces that the value in the Remaining quantity field must always be greater than or equal to zero and less than or equal to the value in the Initial quantity field ($\leq$ Initial quantity)
MET-015	Price Range Violation: Transaction price is zero or out of bounds relative to Limit price.	Rejection	When Status is FILL or PARF, Limit price is populated. If Transaction identification code is also filled, Transaction price value must not be null, must be greater than zero and must be within the range established by the Limit price field. For a BUYI order the Transaction price must not exceed the Limit price, and for a SELL order it must not be below the Limit price.
MET-016	Invalid Enumerated Value for [Field Name]	Rejection	Buy-sell indicator, Validity period, Order type classification, Passive or Aggressive Indicator and Status fields values must match the pre-defined set of values for each field as defined in section 3.4.1 Business fields

Table 47 - Rejection validation rules

7.2.2 Qualitative validation rules

Error code	Error Message	Validation Outcome	Control
QUA-001	Field length limit exceeded for [Field Name]	Warning	Validates maximum character constraints for text fields (e.g. Order identification code is <= 50 characters, Transaction identification code is <= 52 character)
QUA-002	Invalid Execution Venue Market Identification Code: Segment MIC / Trading platform for crypto-asset	Warning	If Transaction identification code is not empty for orderbook records and always for execution records, Segment MIC / Trading platform for crypto-asset must be valid against the global ISO 10383 (except if Segment MIC / Trading platform for crypto-asset = XOFF or XXXX)
QUA-003	[Field Name] is not valid against ISO 24165-2	Warning	ISO 24165 DTIF Registry Check: Price Currency, Crypto-asset identification code field values should be valid against ISO 24165-2.
QUA-004	Extreme Order/Trade Ratio Detected	Warning	The system validates macro ratios across the file: 1. Cancelled orders must be between 5% and 95% of total orders. 2. Order Buy/Sell ratio cannot be extreme (e.g. >90% buy). 3. For trading venues, the Execution Buy/Sell trade count ratio must be within +- 0.1% proximity.
QUA-005	Abnormal Price Discrepancies. [Price] exhibits a variance of 1,000% or greater between two consecutive execution entries under the same [Price Currency]	Warning	Each execution price is compared against its immediately preceding and following executions. A price is flagged where it diverges materially from both neighbors while those neighbors remain consistent with each other. This distinguishes an isolated erroneous print, which departs from the prevailing price and immediately returns, from a genuine price movement, which continues in the same direction across consecutive executions. Partition by Price Currency and Trading platform for crypto-asset.

QUA-006	Segment MIC must be empty when Transaction identification code is not populated.	Warning	If the Transaction identification code field is empty, the Segment MIC field must also be blank/empty.
QUA-007	Conditional required field: Passive or Aggressive Indicator must be empty for the submitted Status.	Warning	The Passive or Aggressive Indicator field can only contain data if the Status is PARF or FILL. For all other transaction/order statuses, this field must be left completely blank/empty.
QUA-008	Abnormal proportion of midnight-defaulted timestamps	Warning	A significant proportion of Trading date time values, or Date and time values for order book records, in the submitted file are set to midnight (00:00:00.000), indicating a likely defaulting issue in the upstream system. The system flags the file for qualitative review when the ratio of midnight-timestamped records exceeds a defined threshold.

Table 48 - Qualitative validation rules

8 Annex 4: JSON file types

Below there is a list of JSON sample files intended to assist CASPs in understanding the expected structure and content of submissions to the MIDAS system:

- Execution (Record Keeping) Sample:

The Execution JSON sample covers trade reporting across three trading pairs, ETH/USDC, HYPE/USDC, and RAY/USDC and includes both buy and sell legs. Multiple execution records share the same Transaction Record Number but carry distinct Trading platform for crypto-asset transaction identification codes, reflecting cases where a single order results in several separate execution reports. Quantities and prices vary across records, providing a realistic representation of the range of values CASPs may report. All timestamps across the file follow the defined YYYY-MM-DD HH:mm:ss.fff format, as required by the MIDAS ingestion rules.

Note that this Sample contains both buy and sell legs, as would be reported by a Trading Venue. A Broker Dealer would only report one of the two legs.



CryptoAssetRecordK
eeping_21052026_par

- Order Book Sample:

The Order Book JSON sample covers a range of key reporting scenarios that CASPs are likely to encounter in practice. The sample demonstrates the full lifecycle of limit orders (LMTO) across two trading pairs, HYPE/USDC and RAY/USDC, starting from order creation (NEWO), progressing through multiple sequential partial fills (PARF), and concluding either with a complete fill (FILL) or a cancellation at the initiative of the participant (CAME). Both buy (BUYI) and sell (SELL) sides are represented throughout. The sample also provides examples of the correct handling of the Passive or Aggressive Indicator field, which is populated with PASV or AGRE on PARF and FILL statuses and left blank on CAME records, reflecting the conditional population rules that apply to this field.



CryptoAssetOrderBo
ok_21052026_part1.js

9 Annex 5: Feedback file



CryptoAssetOrderBook_21052026_part1.js

9.1 Purpose and scope

9.1.1 What the feedback file is

The feedback file is the MIDAS system's response to a submission. It reports the outcome of every validation rule defined in Annex 3 that was triggered by that submission, at both file level and record level, together with the corrective action required of the submitting CASP.

It is the only channel through which MIDAS communicates the result of processing. A record that is rejected is not held in MIDAS and will not appear in any surveillance output. Nothing other than the feedback file will tell a CASP that this has happened.

9.1.2 1.2 What the feedback file is not

- It is **not an acknowledgement of receipt**. Successful transfer of a file to the SFTP endpoint is confirmed by the SFTP server at the point of transfer and is a separate matter from validation.
- It is **not a notification**. No email is sent and no alert is raised when a feedback file is written. It is the CASP's responsibility to retrieve it (see §2).
- It is **not a correction mechanism**. It reports what must be corrected; the correction itself is made at source and submitted in a subsequent file, in accordance with §3.5.4 of the Reporting Instructions.

9.1.3 1.3 Scope

The feedback file covers both report types:

- **Executions (Record Keeping), auth.117** — mandatory for all CASPs.
- **Order book, auth.118** — mandatory only for CASPs operating a trading platform for crypto-assets.

Not every validation rule applies to both report types. The **AppliesTo** column states the scope of each rule, and §5.5 sets out which rules a given CASP can expect to receive.

9.2 Delivery

9.2.1 When the feedback file is generated

MIDAS produces **one feedback file per CASP per day**. It is generated once MIDAS has processed all files available from that CASP in that day's processing cycle, and reports on all of them together.

Feedback is therefore not returned file by file as each submission is processed. A CASP that submits several files receives a single consolidated response covering all of them.

9.2.2 Where the feedback file is written

The feedback file is deposited on the **same SFTP endpoint from which the CASP's submitted files were collected**, in the same root folder in which they deposit their submitted files. CASPs retrieve it using the same connection they use to submit data.

No email is sent and no separate notification is generated. It is the CASP's responsibility to check that folder after each submission. An automated polling job on a fixed interval is recommended in preference to manual checks.

9.2.3 Relationship between submitted files and feedback files

The relationship is **many to one**. A single feedback file covers every file processed for that CASP in that day's cycle, across both report types.

This differs from the one-submission-one-response pattern used by some other reporting systems, and CASPs should account for it when building their handling:

- Submitting six part files for a single reporting date yields one feedback file, not six.
- Submitting an executions file and an order book file on the same day yields one feedback file covering both.
- Every row in the feedback file carries **SubmittedFileName**, which names the submitted file that row's finding was found in. This column is what attributes each finding back to its source file, and it is the only thing that does so.

Consequently a CASP must group findings by **SubmittedFileName** before acting on them.

9.2.4 Naming convention

Every feedback file is named [CASP NAME]_MIDAS_Feedback_File_yyyymmdd.csv, for example KRAKEN_MIDAS_Feedback_File_20260908.csv.

[CASP NAME] is replaced with the submitting CASP's identifier. yyyymmdd is the date on which MIDAS processed the batch, not the date the data relates to.

Because MIDAS produces one feedback file per CASP per processing day, the name carries no reference to any individual submitted file. A CASP that submits six part files on one day receives one feedback file covering all six, and attribution of each finding back to its source file is carried by the SubmittedFileName column rather than by the file name.

9.2.5 Retention

The feedback file remains in that same folder for the same period as the records and orders it relates to, namely seven years.

9.2.6 Format

The feedback file is provided in CSV format, irrespective of the JSON format in which data is submitted. Full specification at §3.1.

9.3 How to read the file

9.3.1 Format

Property	Value
Format	CSV
Encoding	UTF-8
Delimiter	Comma (,)
Quoting	RFC 4180. Any value containing a comma, a double quote or a line break is enclosed in double quotes; embedded double quotes are doubled.
Header	One header row, always present, always the first line
Columns	17, in the fixed order set out in §4
Rows	One row per finding (see §3.2)

9.3.2 The governing principle: one row per finding, not one row per record

Everything else in this section follows from this.

A **finding** is a single validation rule firing against a single subject. The subject may be a record, or it may be the file as a whole. Each finding produces exactly one row.

The consequences are:

- The file reports **issues**. It is not a manifest of the submission and does not carry a row for every record submitted.
- A record that breaches three rules produces **three rows**, all carrying the same **RecordRef** and the same **JSONLineNumber**, and differing in **ErrorCode**.
- Records that breach no rule are not listed individually. A single row reports the total number of records accepted into MIDAS, with Outcome = ACCEPTED_WITHOUT_WARNING.

- A rule evaluated across the whole file produces **one row** with no **RecordRef** and no **JSONLineNumber**.
- A file rejected before processing produces **one row per file-level rule breached**, and no record-level rows at all, because no records were processed.

Therefore the number of rows in the feedback file bears no fixed relationship to the number of records in the submission. Reconciliation is performed on **RecordRef** and **JSONLineNumber**, never on row position.

Because warned records are listed individually and unwarned records are reported only as a total, the submission reconciles exactly:

Total records submitted = the **ACCEPTED_WITHOUT_WARNING** count, plus the number of distinct records carrying **ACCEPTED_WITH_WARNING**, plus the number of distinct records carrying **REJECTED**.

The word distinct is load-bearing. A record breaching three rules produces three rows, so counting rows rather than records would overstate the rejected population. Reconcile on **RecordRef** together with **JSONLineNumber**, never on row count.

9.3.3 The four kinds of row

The columns that are populated differ according to the kind of row. There are four.

Kind	Raised by	RecordRef / JSONLineNumber	Effect on the submission
File-level rejection	Annex 7.1 (FIL codes)	Empty	The whole file is discarded. No records are processed.
Record-level finding	Annex 7.2 (MET and QUA codes)	Populated	Affects that record only. The rest of the file is processed normally.
File-scope content warning	Annex 7.2 (QUA-004, QUA-008)	Empty	Advisory. The file is processed and the records are retained.
Accepted record	No rule triggered	Populated	The record is held in MIDAS.

The third kind deserves particular attention. QUA-004 and QUA-008 are content validation rules, so they carry **FeedbackType = RECORD_CONTENT_VALIDATION**, but they are evaluated against the file as a population rather than against any individual record. They therefore have no record reference, and they are distinguished by **Outcome = FILE_WARNING**.

9.3.4 Reading order

Triage a feedback file in this sequence:

1. **Search for `Outcome = FILE\_REJECTED` first.** If any row carries it, that file was rejected in its entirety and nothing in it was processed. No record-level rows will exist for that file. Correct and resubmit the file in full before considering anything else. You must notify the MIDAS support desk at support@esma.europa.eu when resubmitting, using the header format MIDAS - CASP - <CASP Name> - Topic. This is the same channel used for corrections to previously accepted records under section 3.5.4 of the Reporting Instructions.
2. **Then `Outcome = REJECTED`.** These records are not in MIDAS. Each must be corrected at source and resubmitted.
3. **Then `Outcome = FILE\_WARNING`.** These indicate a systemic problem with the submission as a whole, typically upstream of the file itself. No resubmission is required, but the underlying cause should be investigated.
4. **Then `Outcome = ACCEPTED\_WITH\_WARNING`.** These records are in MIDAS. No resubmission is required. The value should be reviewed at source to prevent recurrence.
5. **`Outcome = ACCEPTED\_WITHOUT\_WARNING` requires no action.**

9.4 Column dictionary

The 17 columns, in file order.

#	Column	Populated on
1	ReportTimestamp	All rows
2	CaspIdentifier	All rows
3	SubmittedFileName	All rows
4	FeedbackType	All rows
5	FileVerdict	All rows
6	RecordRef	Record-level rows only
7	JSONLineNumber	Record-level rows only
8	Outcome	All rows
9	ErrorCode	All rows except accepted records
10	AppliesTo	All rows
11	Label	All rows
12	Severity	All rows
13	Field	Where the finding is attributable to a field
14	LogicalPath	Where the finding is attributable to a field
15	SubmittedValue	Where a value exists to report
16	ValidationReason	All rows
17	CorrectiveAction	All rows

9.4.1 ReportTimestamp

The date and time at which the feedback file was generated, in YYYY-MM-DDThh:mm:ss.fff format, UTC.

The value is identical on every row of a given feedback file. It is a property of the feedback file, not of the individual finding, and it does not indicate when the underlying validation was performed or when the submission was received.

9.4.2 CaspIdentifier

The Legal Entity Identifier of the CASP whose submission is being reported on, as a 20-character LEI conforming to ISO 17442.

Identical on every row of a given feedback file.

9.4.3 SubmittedFileName

The name of the submitted JSON file to which the row relates, exactly as it was received by MIDAS, including the `.json` extension.

This column is the join key between a finding and the submission it came from, and it is the mechanism by which a consolidated daily feedback file (§2.3) attributes each finding to the right submission. Since a feedback file routinely reports on several submitted files, this is the only column that distinguishes them. `RecordRef` and `JSONLineNumber` are scoped to the file named here and are not unique across files.

Note that on FIL-101 rows the value reported is the offending file name itself, which by definition does not conform to the required naming convention.

9.4.4 FeedbackType

The validation layer that produced the finding. See §5.1 for permitted values.

This is the first of four related but non-redundant status columns. Their division of labour is:

Column	Answers
<code>FeedbackType</code>	Which layer of validation ran?
<code>FileVerdict</code>	What happened to the file?
<code>Outcome</code>	What happened to this record?
<code>Severity</code>	How serious is it?

Each is enumerated separately in §5, and §6 sets out which combinations can legitimately occur.

9.4.5 FileVerdict

The disposition of the submitted file named in `SubmittedFileName`. See §5.2.

`REJECTED` means the file failed transmission or file validation and no records within it were processed. `PROCESSED` means the file was accepted for processing; individual records within it may still have been rejected.

`FileVerdict` is a property of the file, so it carries the same value on every row relating to the same `SubmittedFileName`.

9.4.6 RecordRef

The business identifier of the record giving rise to the finding, taken from the record itself as submitted.

The identifier used depends on the report type:

Report	Identifier reported	Logical path
Executions (auth.117)	Transaction Record Number	/Document/CrptAsstRcrdKeepgRpt/Tx/New/TxRcrdId
Order book (auth.118)	Order identification code	/Document/CrptAsstOrdRBookRpt/OrdRpt/New/OrdR/OrdRIdData/OrdRId

Empty on file-level rejection rows and on file-scope content warning rows, neither of which is attributable to a single record.

RecordRef is not necessarily unique within a file. This is a deliberate consequence of the reports' structure and is important to reading the file correctly:

- In the **order book**, every event in an order's lifecycle carries the same Order identification code. A single order that is created, partially filled and then filled produces three records sharing one **RecordRef**. An order-book record is therefore located by **RecordRef** and **JSONLineNumber** together, never by **RecordRef** alone.
- In **executions**, a duplicate record by definition repeats the identifiers of the record it duplicates. Where MET-005 is reported, the same **RecordRef** will appear at more than one **JSONLineNumber**: the first instance, which is retained, and the subsequent instances, which are rejected.

Both cases are illustrated in §8.

9.4.7 JSONLineNumber

The line number within the submitted JSON file named in **SubmittedFileName** at which the record giving rise to the finding is located. Counted from 1 at the first line of the file. Where a record object spans several lines, the value is the opening line of that record object.

Empty on file-level rejection rows and on file-scope content warning rows.

Together with **RecordRef**, this column provides the audit trail from a finding back to the exact record in the exact file the CASP sent, without the CASP needing to search its own data by business key.

Because the value is a physical property of the file as transmitted, it depends on how the file is laid out. A file submitted in pretty-printed form yields a precise and useful pointer. A file submitted in minified form, with the entire JSON document on a single line, will produce **1** on

every row and the column will carry no information. **CASPs are therefore recommended to submit pretty-printed JSON.**

9.4.8 Outcome

The disposition of the subject of the finding — the record where one is identified, otherwise the file. See §5.3.

This is the operative column for deciding what to do. §9 maps each value to the action required.

9.4.9 ErrorCode

The identifier of the validation rule that fired, as published in Annex 3 of the Reporting Instructions. Three families exist:

Prefix	Source	Meaning
FIL-	Annex 7.1, Transmission/file validation rules	File-level. Rejects the entire file.
MET-	Annex 7.2.1, Rejection validation rules	Record-level. Rejects the record.
QUA-	Annex 7.2.2, Qualitative validation rules	Advisory. Does not reject.

ErrorCode is the **stable machine key** of the feedback file. It is the column a CASP's automated handling should switch on. Codes are not reused or renumbered; see §7.5.

9.4.10 AppliesTo

The report type or types to which the validation rule identified in **ErrorCode** applies. See §5.5 for permitted values and the full scope map.

Two properties of this column are easily misread and are stated here explicitly:

It describes the rule, not the row. **AppliesTo = BOTH** does not mean the row relates to both reports; a row always relates to exactly one submitted file. It means the rule is capable of firing against either report type. To determine which report a given row relates to, read **SubmittedFileName** or **LogicalPath**.

It is a property of the error code and is constant. A given **ErrorCode** always carries the same **AppliesTo** value, on every row, in every feedback file. The column is therefore redundant on any individual row, and is included so that a CASP can determine, without consulting Annex 3, which subset of rules its submissions are subject to. §5.5 gives that subset directly.

9.4.11 Label

A stable machine-readable name for the condition that fired, in upper snake case. One label corresponds to exactly one error code, and the mapping is fixed; §5.6 gives it in full.

Label and **Field** are distinct and are frequently confused:

- **`Label`** names the **condition** — what went wrong. **FIELD_MANDATORY_MISSING**, **RECORD_DUPLICATE**, **TXN_ID_MISSING_ON_FILL**.
- **`Field`** names the **business field** — where it went wrong. **Crypto Asset Identification Code**, **Trading date time**, **Limit price**.

A worked comparison across all four descriptive columns, taken from the sample:

Column	Value	What it tells you
ErrorCode	MET-009	Which Annex 3 rule fired
Label	TXN_ID_MISSING_ON_FILL	What the condition is
Field	Transaction identification code	Which business field, as named in §3.4.1
LogicalPath	/Document/CrptAsstOrdRBookRpt/OrdRpt/New/OrdR/OrdRData/TxData/TxId	Where that field physically sits in the JSON

Label is provided so that automated handling can be written against a readable constant rather than a numeric code. Either may be used as the key; they are equivalent.

9.4.12 Severity

The seriousness of the finding, and by extension whether it blocks ingestion. See §5.4.

Severity is closely coupled to **Outcome** but is not the same thing: **Outcome** states what happened to the subject, **Severity** states the class of the rule. The permitted pairings are given in §6.

9.4.13 Field

The name of the business field to which the finding relates, given exactly as that field is named in §3.4.1 of the Reporting Instructions, so that it can be looked up directly.

Three special forms occur:

- **`-`** where the finding is not attributable to any field. FIL-102, which reports a structural failure of the whole document, and accepted-record rows.
- **A collective name** where a rule operates on a composite key rather than on any single field. MET-005 reports **Execution unique key**, the composite that defines duplication, rather than enumerating its constituent fields.

- **Several field names** where a rule operates on more than one field jointly. MET-011 reports **Traded quantity / Transaction price**. See §7.3 for how these align with **LogicalPath** and **SubmittedValue**.

On FIL-101 the value is **SubmittedFileName**, the file name itself being the subject of the rule.

9.4.14 LogicalPath

The location of the field within the JSON document, expressed as the logical path published in §3.4.1 of the Reporting Instructions.

- where no field is implicated, matching **Field**.

Where a rule spans several fields, every applicable path is listed, joined by **+**. The paths are given in the same order as the corresponding entries in **Field** and **SubmittedValue**, and are read positionally. See §7.3.

9.4.15 SubmittedValue

The value that was actually received by MIDAS, reproduced verbatim, including any formatting that contributed to the finding.

This column carries four distinct forms, and the distinction between the first three is load-bearing:

Form	Meaning	Example
A value	The value as submitted	-0.0014159000000000 on MET-004
Empty	The field was absent or null in the submission. On a mandatory-field or conditional-mandatory rule, the emptiness is the finding.	MET-001, MET-007, MET-009, MET-013
-	The column does not apply to this row. No value is being reported.	FIL-102, accepted-record rows
An aggregate	A summary statistic across the file, not a field value. Occurs only on file-scope content warnings.	620 BUYI / 17 SELL of 637 records on QUA-004; 128 of 637 records on QUA-008

An empty cell and a - are therefore **not interchangeable** and must not be normalised to one another when parsing. An empty cell asserts something about the submitted data; a - asserts something about the row.

One further caution. A submitted value may itself be a string that resembles a marker. In the sample, MET-002 reports a **SubmittedValue** of **N/A** — that is the literal three-character string the CASP placed in a numeric price field, and it is the reason the record was rejected. It is data, not a placeholder. Parsers must not treat any particular string as a null sentinel; only a genuinely empty cell signifies absence.

9.4.16 ValidationReason

A plain-language explanation of the finding, written to be read by a person.

Where the rule is conditional, the reason states the condition that was met, so that the finding can be understood without cross-referencing. For example, MET-010 reports not only that a Transaction identification code was present when it should not have been, but also that the submitted Status was **NEWO**, which is what made it impermissible.

Where the rule compares one value against another, the reason states the comparator. MET-012 reports the Initial quantity against which the Traded quantity was found excessive; MET-015 reports the Limit price against which the Transaction price was found out of bounds; MET-008 reports the Date and time against which the Validity period and time was found not to be later.

The text is deterministic for a given error code and set of inputs, but it is **not a stable parsing target**. It is subject to editorial change without a change of error code. Automated handling must key on **ErrorCode** or **Label**.

9.4.17 CorrectiveAction

What the CASP is required to do about the finding, in plain language.

The text is drawn from a fixed set of action patterns, enumerated in §5.7, each of which corresponds to a definite obligation. As with **ValidationReason**, the wording is not a stable parsing target; derive the obligation from **Outcome** and **ErrorCode** per §9.

9.5 Attribute values

9.5.1 FeedbackType

Value	Meaning	Appears with
TRANSMISSION_FILE_VALIDATION	The finding arose from the transmission and file validation layer described in §3.5.1 and enumerated in Annex 7.1.	FIL codes
RECORD_CONTENT_VALIDATION	The finding arose from the content validation layer described in §3.5.2 and enumerated in Annex 7.2.	MET and QUA codes
ACCEPTED_RECORD	The row confirms acceptance of a record that triggered no rule.	No code

9.5.2 FileVerdict

Value	Meaning
REJECTED	The file failed transmission or file validation. No records within it were processed.
PROCESSED	The file was accepted for processing. Individual records may still have been rejected.

9.5.3 Outcome

Value	Subject	Meaning	In MIDAS?
FILE_REJECTED	The file	The file was rejected before any record was processed.	No records
REJECTED	A record	The record failed a rejection rule and was not ingested.	No
ACCEPTED_WITH_WARNING	A record	The record was ingested, but a qualitative rule flagged it for review.	Yes
FILE_WARNING	The file	A qualitative rule evaluated across the whole file was breached. No individual record is implicated.	Yes, all records
ACCEPTED_WITHOUT_WARNING	The submission	The total number of records that triggered no rule and were accepted into MIDAS. Reported once as an aggregate, not once per record.	Yes

9.5.4 Severity

Value	Meaning	Blocks ingestion?
FILE_REJECTION	A file-level failure. The submission cannot be processed at all.	Yes, the whole file
REJECTION	A record-level failure.	Yes, that record
WARNING	A data quality observation.	No
INFO	Confirmation of acceptance. Not a finding.	No

9.5.5 AppliesTo

Value	Meaning
EXECUTION	The rule applies only to Executions (Record Keeping) submissions, auth.117.
ORDERBOOK	The rule applies only to order book submissions, auth.118.
BOTH	The rule applies to both report types.

The scope of every published rule:

Scope	Count	Codes
BOTH	12	FIL-101, FIL-102, MET-001, MET-002, MET-003, MET-004, MET-005, MET-006, QUA-002, QUA-003, QUA-004, QUA-008
ORDERBOOK	13	MET-007, MET-008, MET-009, MET-010, MET-011, MET-012, MET-013, MET-014, MET-015, MET-016, QUA-001, QUA-006, QUA-007
EXECUTION	1	QUA-005

9.5.6 ErrorCode and Label

The mapping is one-to-one and fixed. Either may be used as the key for automated handling.

Code	Label	Applies to	Severity
FIL-101	FILE_NAMING_CONVENTION	BOTH	FILE_REJECTION
FIL-102	FILE_SCHEMA_CONFORMANCE	BOTH	FILE_REJECTION
MET-001	FIELD_MANDATORY_MISSING	BOTH	REJECTION
MET-002	FIELD_TYPE_INVALID	BOTH	REJECTION
MET-003	FIELD_VALUE_OUT_OF_RANGE	BOTH	REJECTION
MET-004	FIELD_VALUE_NOT_POSITIVE	BOTH	REJECTION
MET-005	RECORD_DUPLICATE	BOTH	REJECTION
MET-006	TIMESTAMP_FORMAT_MISMATCH	BOTH	REJECTION
MET-007	GTDV_VALIDITY_TIME_MISSING	ORDERBOOK	REJECTION
MET-008	VALIDITY_TIME_NOT_AFTER_DATE_TIME	ORDERBOOK	REJECTION
MET-009	TXN_ID_MISSING_ON_FILL	ORDERBOOK	REJECTION
MET-010	TXN_ID_PRESENT_WITHOUT_FILL	ORDERBOOK	REJECTION
MET-011	EXEC_DETAIL_WITHOUT_TXN_ID	ORDERBOOK	REJECTION
MET-012	TRADED_QTY_EXCEEDS_INITIAL	ORDERBOOK	REJECTION
MET-013	LIMIT_ORDER_PRICE_MISSING	ORDERBOOK	REJECTION
MET-014	REMAINING_QTY_OUT_OF_BOUNDS	ORDERBOOK	REJECTION
MET-015	TXN_PRICE_OUTSIDE_LIMIT	ORDERBOOK	REJECTION
MET-016	ENUM_VALUE_INVALID	ORDERBOOK	REJECTION
QUA-001	FIELD_LENGTH_EXCEEDED	ORDERBOOK	WARNING
QUA-002	MIC_NOT_IN_ISO10383	BOTH	WARNING
QUA-003	DTI_NOT_IN_ISO24165	BOTH	WARNING
QUA-004	MACRO_RATIO_ANOMALY	BOTH	WARNING
QUA-005	PRICE_OUTLIER_ANOMALY	EXECUTION	WARNING
QUA-006	SEGMENT_MIC_WITHOUT_TXN_ID	ORDERBOOK	WARNING
QUA-007	PASV_AGRE_INVALID_FOR_STATUS	ORDERBOOK	WARNING
QUA-008	MIDNIGHT_TIMESTAMP_ANOMALY	BOTH	WARNING
-	RECORDS_ACCEPTED	BOTH	INFO

9.5.7 CorrectiveAction patterns

Pattern	Obligation	Applies to
Correct the file name and resubmit the file in full.	Resubmit the entire file under a conforming name.	FIL-101

Correct the file structure and resubmit the file in full.	Resubmit the entire file with conforming structure.	FIL-102
Correct the record at source and resubmit only the corrected record in the next submission.	Fix at source; resubmit that record only. Do not resubmit the whole file.	MET-001 to MET-004, MET-006 to MET-010, MET-012 to MET-016
Review the reason why these fields are populated for records that are not fills or partial fills. Correct at source and resubmit only the corrected record in the next submission.	As above, with a diagnostic pointer to the likely upstream cause.	MET-011
No resubmission of this record required. Remove the duplicate at source to prevent recurrence.	Do not resubmit. The retained instance is already in MIDAS. Fix the process that emitted the duplicate.	MET-005
No resubmission required. Review the value at source to prevent recurrence.	Record is in MIDAS. Correct the source data so future submissions are clean.	QUA-001, QUA-002, QUA-003, QUA-006, QUA-007
No resubmission required. Investigate whether both trade legs are being captured upstream.	Investigate the completeness of the upstream feed.	QUA-004
No resubmission required. Investigate the upstream price source.	Investigate the price feed.	QUA-005
No resubmission required. Investigate the upstream timestamp source.	Investigate timestamp defaulting upstream.	QUA-008
None required.	No action.	Accepted records

Note that MET-005 is the one rejection that does **not** call for resubmission. A duplicate is rejected precisely because the record it duplicates was accepted; resubmitting it would produce a further duplicate.

9.6 Legal combinations

Five row shapes occur. No other combination of the status columns is valid, and a parser may assert against this table.

Shape	FeedbackType	FileVerdict	Outcome	Severity	RecordRef / JSONLine Number	Error Code
1. File rejected	TRANSMISSION_FILE_VALIDATION	REJECTED	FILE_REJECTED	FILE_REJECTION	Empty	FIL
2. Record rejected	RECORD_CONTENT_VALIDATION	PROCESSED	REJECTED	REJECTION	Populated	MET

3. Record warned	RECORD_CONTENT_VALIDATION	PROCESSED	ACCEPTED_WITH_WARNING	WARNING	Populated	QUA
4. File warned	RECORD_CONTENT_VALIDATION	PROCESSED	FILE_WARNING	WARNING	Empty	QUA
5. Record accepted	ACCEPTED_RECORD	PROCESSED	ACCEPTED_WITHOUT_WARNING	INFO	Populated	-

Two properties worth noting:

- **Shapes 1 and 2 are mutually exclusive within a file.** If a file yields shape 1, it cannot yield shapes 2 to 5, because no records were processed.
- **`AppliesTo` is independent of all five shapes.** It is determined solely by **ErrorCode** and constrains none of the status columns.

9.7 Conventions

9.7.1 Empty, -, and literal values

Three states must be distinguished and must not be normalised to one another:

State	Assertion
Empty cell	About the data . The field was absent or null in the submission. On mandatory and conditional-mandatory rules this is itself the finding.
-	About the row . The column is not applicable to this kind of row.
Any other string	The literal content of the field as submitted, including strings such as N/A that may resemble placeholders.

9.7.2 Timestamps

All date and time values in the feedback file are given as **YYYY-MM-DDThh:mm:ss.fff** in UTC. Values reproduced in **SubmittedValue** and **ValidationReason** are given as submitted and may not conform, since non-conformance is frequently the finding being reported.

9.7.3 Composite rules

Where a validation rule operates on more than one field, **Field**, **LogicalPath** and **SubmittedValue** each list the corresponding elements, in the same order. Read them positionally: the *n*th logical path corresponds to the *n*th field name and the *n*th submitted value.

`LogicalPath` joins its elements with `+`. `Field` and `SubmittedValue` use the separator shown in the row.

Where a rule operates on a composite key as a single concept rather than on its constituent fields individually, `Field` gives a collective name for the key. MET-005 is the example: `Field` reads `Execution unique key`, while `LogicalPath` and `SubmittedValue` enumerate the four elements that make up the key.

9.7.4 Row order

Rows are presented with file-level findings first, then grouped by submitted file and ordered by error code.

Row order is presentational and must not be relied upon by automated handling. Use `ErrorCode`, `SubmittedFileName`, `RecordRef` and `JSONLineNumber` to locate and group findings.

9.7.5 Stability

The following are a stable contract and will not change without a versioned revision of these instructions:

- The set and order of columns.
- The `ErrorCode` values and their meanings. Codes are not reused or renumbered.
- The `Label` values and their one-to-one mapping to error codes.
- The permitted values of `FeedbackType`, `FileVerdict`, `Outcome`, `Severity` and `AppliesTo`.

The following are **not** stable and must not be parsed:

- The wording of `ValidationReason` and `CorrectiveAction`.
- Row order.

9.8 Worked walkthrough of the sample

The accompanying sample, `MIDAS_Feedback_File_Sample.csv`, is a single feedback file covering four submitted files from one CASP. It exercises every published validation rule. Each case below refers to rows of that file.

9.8.1 A file rejected before processing

Two rows, both shape 1, reporting two separate submitted files.

`CryptoAssetRecordKeeping_06082026_1_1.csv` breaches FIL-101. The naming convention mandates the `.json` extension and MIDAS accepts JSON submissions only, so a CSV file is rejected on its name before its contents are examined. The required format is set out in §3.3; the same file submitted as `CryptoAssetRecordKeeping_06082026_1_1.json` would have been accepted for processing. `Field` reads `SubmittedFileName` and `SubmittedValue` reproduces the offending name.

`CryptoAssetRecordKeeping_06082026_2_2.json` breaches **FIL-102**, a JSON schema conformance failure. Because the failure is structural and pertains to the document as a whole, `Field`, `LogicalPath` and `SubmittedValue` all read -.

In both cases `FileVerdict` is **REJECTED** and no record-level rows appear for those files anywhere in the feedback file. Both require resubmission in full.

9.8.2 A processed file with rejected records

`CryptoAssetRecordKeeping_06082026_1_1.json` carries `FileVerdict` = **PROCESSED** throughout. Individual records within it were nonetheless rejected.

Take the **MET-001** row and follow it end to end:

Column	Value
<code>SubmittedFileName</code>	<code>CryptoAssetRecordKeeping_06082026_1_1.json</code>
<code>RecordRef</code>	<code>1779321603054903072</code>
<code>JSONLineNumber</code>	<code>2</code>
<code>Outcome</code>	REJECTED
<code>ErrorCode</code>	MET-001
<code>Label</code>	FIELD_MANDATORY_MISSING
<code>Field</code>	<code>Crypto Asset Identification Code</code>
<code>LogicalPath</code>	<code>/Document/CrptAsstRcrdKeepgRpt/Tx/New/CrptAsst/DgtITknldr/DTI</code>
<code>SubmittedValue</code>	<i>(empty)</i>

The CASP opens the named file, goes to line 2, confirms the record's Transaction Record Number is `1779321603054903072`, and finds the **DTI** element null. The empty `SubmittedValue` is not a gap in the feedback, it is the finding. The record is not in MIDAS and must be corrected at source and resubmitted on its own.

9.8.3 A record accepted but carrying a warning

The **QUA-002** row reports a Trading platform for crypto-asset of **PGSX**, which does not resolve against ISO 10383. `Outcome` is **ACCEPTED_WITH_WARNING** and `Severity` is **WARNING**.

The distinction from §8.2 is material: this record **is** in MIDAS and is available to surveillance. No resubmission is required. The obligation is to correct the source so that later submissions carry a valid MIC.

9.8.4 A whole-file quality warning with no record reference

Two rows, both shape 4, and the only rows in the file where a content validation carries no record reference.

QUA-004 reports `620 BUYI / 17 SELL of 637 records` in `SubmittedValue`, an aggregate, not a field value. The execution buy/sell ratio of 97.3% to 2.7% suggests that only one leg of each trade is

reaching MIDAS. **RecordRef** and **JSONLineNumber** are empty because no individual record is at fault; the population is.

QUA-008 reports **128 of 637 records**, 20.1% of Trading date time values falling exactly on midnight, indicating timestamp defaulting upstream.

Neither requires resubmission. Both point at a systemic problem that will recur in every future submission until it is fixed at source, which is why the corrective action directs investigation rather than correction of the file.

9.8.5 An accepted record

One row, shape 5. FeedbackType is ACCEPTED_RECORD, Outcome is ACCEPTED_WITHOUT_WARNING and Severity is INFO. RecordRef and JSONLineNumber are empty because the row is an aggregate rather than a reference to one record, ErrorCode, Field and LogicalPath read - because no rule fired and no field is implicated, and SubmittedValue carries the count of records accepted.

This row exists because §3.5.3 requires the feedback file to confirm the acceptance of records that triggered no validation rule.

The feedback file is **issue-driven**: it is not a per-record manifest of the submission. It does not carry one row per accepted record, and a CASP submitting 100,000 records of which three are defective should not expect 100,000 rows. Acceptance is confirmed on the basis set out above; absence of a finding against a record is itself confirmation that the record was accepted.

9.8.6 One record breaching several rules

Ord_1779321887089541910 at line 1 of **CryptoAssetOrderBook_06082026_1_1.json** produces **eight** separate rows:

ErrorCode	Label	Field
MET-003	FIELD_VALUE_OUT_OF_RANGE	Date and time
MET-007	GTDV_VALIDITY_TIME_MISSING	Validity period and time
MET-008	VALIDITY_TIME_NOT_AFTER_DATE_TIME	Validity period and time
MET-010	TXN_ID_PRESENT_WITHOUT_FILL	Transaction identification code
MET-011	EXEC_DETAIL_WITHOUT_TXN_ID	Traded quantity / Transaction price
MET-016	ENUM_VALUE_INVALID	Buy-sell indicator
QUA-006	SEGMENT_MIC_WITHOUT_TXN_ID	Segment MIC
QUA-007	PASV_AGRE_INVALID_FOR_STATUS	Passive or aggressive indicator

All eight share the same **RecordRef** and **JSONLineNumber**. Six are rejections and two are warnings, and the record is rejected on the strength of the six. Note also that two rows, MET-007 and MET-008, implicate the same field for different reasons.

A CASP correcting this record must address all eight findings before resubmitting. Fixing one and resubmitting will simply produce seven findings on the next cycle.

9.8.7 Locating a record where **RecordRef** repeats

Two cases in the sample show why **RecordRef** alone is not sufficient.

Order lifecycle. **Ord_1779321887089541910** appears at lines 1, 2 and 32 of the order book file. These are three separate records, successive events in the life of one order, all carrying the same Order identification code. Findings at line 2 (MET-009, MET-012, MET-014, MET-015) relate to a different record from those at line 1 in §8.6.

Duplicates. In the executions file, **RecordRef 1779321603054903058** appears at line 4 carrying **MET-005**, and also at line 1. Line 4 is the duplicate and was rejected; line 1 is where the first instance sits. This is the expected shape of a duplicate finding, and it is the reason MET-005 does not call for resubmission, the record's content is already in MIDAS, submitted at line 1.

9.8.8 Coverage of the sample

The sample exercises all 26 published validation rules, FIL-101 and FIL-102, MET-001 through MET-016, and QUA-001 through QUA-008, together with one accepted-record row. It is intended to give CASPs a concrete reference against which to build and test their handling of the feedback file. It is illustrative: the values in it are constructed, and it does not represent the volume or distribution of findings that a real submission would produce.

9.9 Acting on the feedback file

9.9.1 Deriving the resubmission set

Read **Outcome** first, then qualify by **ErrorCode**:

Outcome	Action	Resubmit
FILE_REJECTED	Correct the file name or structure.	The entire file
REJECTED	Correct the record at source.	That record only, in the next submission
REJECTED and ErrorCode = MET-005	Remove the duplicate at source.	Nothing. The record is already in MIDAS.
ACCEPTED_WITH_WARNING	Correct at source to prevent recurrence.	Nothing
FILE_WARNING	Investigate the upstream cause.	Nothing
ACCEPTED_WITHOUT_WARNING	None.	Nothing

Warnings never require resubmission. Resubmitting a record that was accepted with a warning will cause it to be rejected as a duplicate.

Where a file yields **FILE_REJECTED**, do not look for record-level rows: there are none, because no records were processed. The whole file must be corrected and resent.

Full requirements for corrections, amendments to previously accepted records, and resubmission of unchanged files are set out in §3.5.4 of the Reporting Instructions and are not repeated here.

9.9.2 Automated handling

CASPs intending to process the feedback file programmatically should observe the following:

- Key on **`ErrorCode`** or **`Label`**. Both are stable. Never key on **ValidationReason** or **CorrectiveAction**.
- Determine the action from **`Outcome`**, qualified by **ErrorCode** for the MET-005 case above.
- Group findings by **`SubmittedFileName` + `RecordRef` + `JSONLineNumber`** to obtain the set of findings per record. Grouping by **RecordRef** alone is incorrect for order book submissions, where lifecycle events share an order identification code.
- Treat an empty **SubmittedValue** and a - as different values (see §7.1).
- Do not rely on row order (see §7.4).
- Assert incoming rows against the five legal shapes in §6; any other combination indicates a parsing error.