

IEF Format Specification

Table of Contents

1. Introduction	1
1.1. Version history	1
1.2. Intended audience	1
1.3. IIP, IEF and IUCLID	2
1.4. Usage of the IIP Exchange Format	2
1.5. How to get started quickly?	3
1.6. IUCLID entity types & features not supported in IEF	3
1.7. IUCLID features not supported in the current IIP processing of the IEF format	4
1.8. IEF features available in the format, but not implemented in IIP processing	4
2. The IEF file structure	5
2.1. A simple example	5
3. High-level overview of the IEF file structure	6
3.1. IEF Header	6
3.1.1. VERSION	6
3.1.2. TEMPLATE_VERSION	6
3.1.3. IUCLID_VERSION	7
3.1.4. DATE	7
3.1.5. PARENT	7
3.2. IEF_CUSTOM	8
3.3. Document section	8
3.3.1. TYPE definition	9
3.3.2. Field definition	9
The NAME column	10
Mandatory field identifier	11
3.3.3. Field data section with one or many instances	11
3.3.4. Field data for one document instance	11
3.4. End marker	13
4. IEF syntax	14
4.1. File suffix	14
4.2. Encoding	14
4.3. Line breaks	14
4.4. Comments	14
4.5. Cell Delimiter	14
4.6. Quoting cell values, masking of the quote character	14
4.7. Intra-cell delimiter	15
4.8. Line separator in text cells	15
5. Some IUCLID basics	16
5.1. IUCLID document types	16

5.2. IUCLID field types	16
5.3. Field identifier	17
5.3.1. Abbreviated field identifier	17
5.4. Field types and display in the IUCLID web frontend	18
5.5. Phrasegroups	19
5.6. Blocks and references to blocks	19
6. Mapping of IIP values to IUCLID field types	21
6.1. Boolean	21
6.2. Date	21
6.3. Numeric	22
6.4. Text	23
6.5. Picklist	23
6.6. Quantity	25
6.6.1. Picklist matching	25
6.7. Range	26
6.7.1. Half-bounded Range	26
6.7.2. Full Range	27
6.8. Document Reference	28
6.9. Block	31
6.10. Block Reference	31
6.11. Inventory Reference	32
6.12. Attachment	32
6.13. Data Protection	33
6.14. Field type with multiple values	34
6.15. Preview: Specific mapping to avoid special characters	34

Chapter 1. Introduction

This document is a technical format description for the **IIP Exchange Format (IEF)** in its version 1.1.

The IEF is an exchange format for the [IUCLID Integration Platform \(IIP\)](#), a software provided by [CropLife Europe](#).

It is currently available in a CSV-type format (IEF CSV) and in an XML (IEF XML) representation. This manual is primarily about the CSV-type format specifically.

For the XML representation, the available XSD exists as guidance. For the syntax of field content, this manual is also valid for the XML representation, especially section [Mapping of IIP values to IUCLID field types](#). At a later stage, the structure of this manual will be changed to reflect the differences and similarities between the two import representations.

1.1. Version history

Date	Document version	IEF syntax version	Comment
Jul 17 2024	1.4	1.1.	Separation of unit from range and quantity, leading to IEF version 1.1.
Mar 23 2024	1.3	1.0	Added picklist matching explanation, with partial substring matching
Jul 07 2023	1.3	1.0	Added mapping example values and data fields for IIP version 1.3.0 (no IEF format change)
Jun 11 2023	1.1	1.0	Updated description for IIP version 1.3.0 (no IEF format change)
Jan 21 2023	1.0	1.0	First release, for the IEF version 1.0

1.2. Intended audience

This document is aimed at data architects and IT-professional that know the features of the IUCLID Integration Platform and want to understand the syntax of IEF files, in order to generate IEF files from own data sources. A basic understanding of IUCLID and its information concepts is required.

1.3. IIP, IEF and IUCLID

The IIP tool can generate IEF templates (files with a predefined structure, where data can be added) and IEF data files from a connected IUCLID instance. Inversly, it can import IEF data files and can import the data into the connected IUCLID instance. The IIP tool can work with current and future IUCLID releases and new IUCLID formats without additional changes, as long as the IUCLID format definition syntax and the API (Application Programming Interface) to IUCLID does not change.

The IEF templates and data files represent the IUCLID format as defined in the connected IUCLID instance, with the latest version at the time of writing IUCLID6 8.x. When a new major IUCLID version is released, the IEF files should be generated according to that latest version. Existing IEF files may fail to be processable with newer major versions, if the definition of the respective format is changed in a way, that existing fields do no longer exist.

This document deals only with the IEF format as such. For information about how to generate / process IEF files in the IIP and the details of the processing please check the information about the IIP.

1.4. Usage of the IIP Exchange Format

The IEF can be used to

- transfer information from inhouse systems to IUCLID via the IIP
- exchange information between different IUCLID instances via the IIP

Its purpose is to be generated automatically by report generators from inhouse source systems, in order to transfer complex and / or repetitive information into IUCLID. For simpler data, the data can also be entered using the IIP graphical user frontend, which contains feautres to enter information more efficiently than in the native IUCLID interface.

The IIP software can generate IEF templates for all IUCLID document types that can be used to develop the IIP data generation tools. Also, any existing IUCLID document with any data can be exported in the IEF, directly clarifying details about the required data generation process.



It is not advised to generate or edit the IEF manually with a text editor, it is not designed to be human-readable. Also, Microsoft Excel should not be used to open or edit IEF files, as Excel automatically applies format transformation based on own rules, turning the IEF invalid.

What are the differences of the IEF compared to the I6Z format:

- The IEF allows to supply field information in a simple form that can be generated by source systems without prior knowledge of IUCLID, e.g.
 - specify ranges in one string, e.g. "< 15 < 33 mg/l" instead of multiple fields (here 5)
 - specify picklist values in clear text (instead of IUCLID internal code values)
 - specify references using business values of the target types (e.g. supplying the target document name, CAS number, Study No.)

- The IEF represents tabular information in a tabular way, so large quantities of repetitive field data can be uploaded efficiently
- IEF supports the exchange and comparison of document information between different IUCLID instances, using business values instead of technical identifier.
- The I6Z format relies on the UUID as technical identifiers, the IIP uses business values to detect potential duplicates, e.g. CAS-Numbers or Study Numbers. This allows
 - to export the IEF from applicant data sources, where IUCLID UUIDs are unknown
 - to exchange and merge information between different IUCLID instances, where information was created independently
- An IIP file can contain one to many "jobs", each representing the creation or update of an IUCLID entity or document type. For each job a different and separate job execution type can be previewed and applied, e.g. dependent on the data in the target instance. This is cumbersome to do with an I6Z file and the "Advanced import" in IUCLID

The IEF was developed to support efficient data upload for information in section documents inside of existing substance, mixture or template dataset. In theory you could populate an empty substance dataset with IIP data alone, if all the required information is in the IIP files.

However, the IEF does not intend to replace the I6Z format.

- When generated from a selected mixture, substance or template, the I6Z file will contain all entities and documents that are being referenced recursively, so all referenced test materials, reference substance, and literature references will be part of the I6Z. When importing, this complete graph of documents / entities will be recreated. This is the same when creating a dossier.
- When generating an IIP file from a selected entity, the file will only contain the documents directly in this entity and will not follow any further references to other entities, as the IIP focuses on the efficient creation / update of instances of individual documents, not of a complete graph of interrelated documents of different types.

1.5. How to get started quickly?

If you are impatient and like to try out things, you can also generate IEF files from existing IUCLID documents with the export function of the IIP. By this way you can quickly learn the format and its syntax by looking at those examples and use this document as a reference.

1.6. IUCLID entity types & features not supported in IEF

- IUCLID document types:
 - Contact: no display and management, but supported as target of a reference
 - Legal entity: no display and management, but supported as target of a reference
 - Site: no display and management, but supported as target of a reference

- Article: for the SCIP notifications - substances of concern
- Category: to classify and organize chemical substances
- User / User Roles
- Some fields in IUCLID are defined as multilingual fields. The IEF only considers the "default". Currently, in IUCLID6 only the Poison Centre Notification dossiers use multilingual content
- Export / import of annotation information. A custom annotations are used to tag newly created documents in the Demo mode (see IIP installation manual)

1.7. IUCLID features not supported in the current IIP processing of the IEF format

- required / mandatory attributes
The IIP does not check an IEF file for required / mandatory attributes according to the IUCLID format definitions

1.8. IEF features available in the format, but not implemented in IIP processing

- As of now, the IIP does replace existing IUCLID documents with the data found in the IEF file. There is no mode to merge / append information from the IEF file into the existing IUCLID document.
- External files for rich text fields
The IEF format defines the possibility to use external files for rich text format content, which are referenced from the IEF file. This feature is not yet implemented.
- The values in the marker PARENT column are currently not evaluated.
- The values in the marker IEF_ORIGINID column are currently not evaluated.
- The values in the marker IEF_IUCLIDID column are currently not evaluated.

Chapter 2. The IEF file structure

The IEF is represented as a file, that can be processed by the IUCLID Integration Platform. This chapter describes the internal structure of an IEF file.

2.1. A simple example

For a better initial understanding, here is a very simple IIP file, setting the temperature field in the Results and Discussion Section in a document of type ENDPOINT_STUDY_RECORD.Density.

Listing 1. IIP example

```
##Generated at 2022-12-19T17:12:32.917902700Z  
VERSION;IIP 1.1.2 ①  
TEMPLATE_VERSION;1.0 ②  
IUCLID_VERSION;6.27.2 ③  
DATE;DD/MM/YYYY ④  
TYPE;"ENDPOINT_STUDY_RECORD.Density" ⑤  
NAME;"ResultsAndDiscussion.Density.Temp" ⑥  
"Quantity";"123,34 °C" ⑦  
END ⑧
```

The individual lines in the IEF file have the following meaning:

- ① version with which the data (or the IIP template) was generated
- ② version of the IEF, here "1.0"
- ③ IUCLID version based on which the template was generated
- ④ specification of the date format
- ⑤ specification of the document type for which data is supplied
- ⑥ Document type header line Definition of the fields that are supplied further below
- ⑦ The values provided for the fields defined in the header line
- ⑧ End marker of the IIP file

The above example uses the semicolon ";" sign as delimiter; this is implicitly derived from the IEF header line with the VERSION marker. Other delimiters are supported. Consequently the IEF is a type of CSV file type; with different values separated by a delimiter.

When the IEF file is processed by the IIP software, the file is parsed and after successful parsing the values are mapped to the respective IUCLID fields. Details of the mapping are explained in the chapter [Mapping of IIP values to IUCLID field types](#)

Chapter 3. High-level overview of the IEF file structure

An IEF file has the following structural elements

- IEF Header
- document section(s)
- End marker

The following chapter describe those elements in more detail:

3.1. IEF Header

An IEF header has the following defined elements, consisting of a marker, followed by the delimiter and the respective value (see the example below)

3.1.1. VERSION

Marker name	VERSION
Mandatory	Yes
Explanation	Version of the software with which the content was created. For the IIP, this will be the IIP version; for applicant tools the tool version should be added here.
Purpose	Tracing and debugging in case of issues
Example	IIP 1.2.0

3.1.2. TEMPLATE_VERSION

Marker name	TEMPLATE_VERSION
Mandatory	Yes
Explanation	The current format version of the IEF. The current value is 1.0. Please note that the IEF format is independent on the IUCLID format - a new IUCLID format does not require a new IEF format, but eventually a new template version.
Purpose	Future check by the IIP, for example for backwards compatibility
Example	1.0

3.1.3. IUCLID_VERSION

Marker name	IUCLID_VERSION
Mandatory	Yes
Explanation	String of the IUCLID6 version, based on which the IEF format was generated, as template or content file. This is generated by the IIP software and should be kept when using a template generated by the IIP. Currently, it is informative only. A warning will be raised during import when the values in this marker differs from the version of the connected IUCLID instance. During job execution errors may occur, when the IEF format cannot be mapped on the IUCLID format of the connected instance.
Purpose	
Example	6.27.2 (Please note that this string does not include the "6" of the IUCLID6 family version, the full version is actually "6.6.27.2". However this family information cannot be retrieved from IUCLID6.

3.1.4. DATE

Marker name	DATE
Mandatory	Yes
Explanation	Format of the date fields supplied in the IEF format. Supported formats are: YYYY-MM-DD, DD-MM-YYYY, MM-DD-YYYY, DD.MM.YYYY, MM/DD/YYYY, DD/MM/YYYY
Purpose	Required for correct parsing of the IEF file by the IIP
Example	YYYY-MM-DD

3.1.5. PARENT

Marker name	PARENT
Mandatory	No
Explanation	For automatic assignment of section documents to a parent entity. This marker supplies information to identify the parent entity with business values.

<p>Purpose</p>	<p>If the IEF data sections only contain section documents, then information about the parent entity can be provided, where the section documents have been exported from / should be imported to. The information is divided in multiple columns:</p> <ul style="list-style-type: none"> • Parent entity type (MIXTURE, SUBSTANCE, TEMPLATE) • Search parameter for the parent entity, at least one out of <ul style="list-style-type: none"> ◦ UUID ◦ name of entity ◦ For SUBSTANCE and MIXTURE only: Other identifier, as given in the Name column of the table "Other identifiers", prefixed with "other:". Example "other:AlternativeName" <p>The search is done using the values from left to right; the subsequent search parameters are only used, if the preceding values did not return any match</p>
<p>Example</p>	<p>MIXTURE;"TestParentMixture"</p>



The marker PARENT is currently not evaluated by the IIP version 1.3.0. The marker is specified for future use.

3.2. IEF_CUSTOM

<p>Marker name</p>	<p>IEF_CUSTOM</p>
<p>Mandatory</p>	<p>No</p>
<p>Explanation</p>	<p>This allows the applicant to add additional header information. This information is ignored by the IIP and not processed.</p>
<p>Purpose</p>	<p>For applicant inhouse use</p>

3.3. Document section

A document section contains of three sub elements. Each document section defines field definition information and content for one to many instances of **one** specific IUCLID document type definition. It has the following substructure:

- TYPE definition

- Field definition
- Field data section(s)

There can be one or many document sections in one IEF file:

- If only section documents are to be processed, the IEF file can contain one to many document sections with different IUCLID document types defined. Consequently, the content of all section documents in a dataset can be stored in one IIF file if needed.
- For simple and complex entites, you can define only one document section in one IEF file. E.g. you cannot import both Literature Reference data and section document data in one IEF file.

3.3.1. TYPE definition

The **TYPE** definition specifies the type of the IUCLID document for which the subsequent field definitions and field data section(s) belong to. The information consists of one line in the IEF file with the following cells (from left to right), separated by delimiter:

1. The marker **TYPE** (mandatory)
2. The technical document name from IUCLID (mandatory).
Examples:
 - **ENDPOINT_STUDY_RECORD.ResiduesInRotationalCrops**
 - **MIXTURE**
 - **LITERATURE_REFERENCE**
3. Version of the document definition, as provided by IUCLID (optional)
4. Provider of the document definition, as provided by IUCLID (optional)

Listing 2. Example of TYPE definition

```
"Type";"ENDPOINT_STUDY_RECORD.AdsorptionDesorption";"7.0";"oecd"
```

3.3.2. Field definition

The field definition defines the fields for which data is going to be provided in the subsequent data sections. The information is provided on a single line. The first 4 columns from left to right are fixed, from column 5 onwards the column header are specific for the TYPE:

1. **NAME** for the column providing the name of the document (column mandatory).
2. **IEF_ORIGINID** for an identifier of the source or origin of a document in e.g. an inhouse system (column optional). This column will be imported as an annotation to allow tracking of source identifiers within IUCLID.



The column IEF_ORIGINID is currently not evaluated by the IIP version 1.3.0. The column is specified for future use.

1. **IEF_IUCLIDID** for an UUID identifier of the IUCLID document (column optional). This information

is used to directly identify possible update candidates. If the UUID matches an existing document within the target IUCLID instance, it uniquely identifies the update candidate. For an update of documents in datasets, the UUID must be present in the set of documents from the dataset or inherited templates in order to be found.



The column IEF_IUCLIDID is currently not evaluated by the IIP version 1.3.0. The column is specified for future use.

1. **IEF_COMMENT** for an informative field, that is not processed by the IIP (column optional). This column can be used to provide additional information for dedicated lines in the field data sections.
2. From column 5 onwards: abbreviated field identifier for the document type defined by **TYPE**. In a full template, the abbreviated field identifier from left to right match the IUCLID field identifier in the document definition or display read from top to bottom; hence the definition is rotated in the IEF by 90 degrees counterclockwards with respect to IUCLID. In a similar manner, also the data in the following field data section will be rotated by 90 degrees.

The following example is a field definition for two fields of the document type **ENDPOINT_STUDY_RECORD.AdsorptionDesorption**

```
NAME;"IEF_ORIGINID";"IEF_IUCLIDID";"IEF_COMMENT";"AdministrativeData.Endpoint";"AdministrativeData.StudyResultType"
```

In the subsequent field data sections, information about exactly those fields can be provided.

For each IUCLID field type there is one identifier, even if in the IUCLID interface the field consists of multiple controls. In IEF each IUCLID field is represented by one cell. Example: A quantity in IUCLID consists of a value field and a picklist with a unit, in IUEF this information is expected as one cell, e.g. **152 mg/l**

The NAME column

The mandatory IEF column with the header **NAME** contains information which is mapped to IUCLID dependent on the type of the target document:

- For section documents this is the name of the document displayed in the top line of the IUCLID user interface. Please note that the name is **not** part of the IUCLID format definition, but supplied in addition. The name is not a mandatory attribute in IUCLID, it is allowed to save section documents without a value in the name.
- For **LITERATURE**, the **NAME** column is the value of the field **GeneralInfo.CompanyOwnerStudyNo**
- For **REFERENCE_SUBSTANCE**, the **NAME** column is the value of the field **ReferenceSubstanceName**
- For **TEST_MATERIAL_INFORMATION**, the **NAME** column is the value of the field **Name**
- For all other entities (**SUBSTANCE,MIXTURE,TEMPLATE**) the **NAME** column is the value of the field **Name**

The specific token **IEF_EMPTY** can be used during IEF import to explicitly specify an empty name value. This token is also used during export.

Mandatory field identifier

From column 5 onwards, abbreviated IUCLID field identifiers are provided. The IEF templates generated by the IIP will contain the complete list of abbreviated field identifiers. In addition, block identifiers are required as field identifiers for the target definitions of "blockref" fields.

When an IEF template is filled with data, those field identifiers for which no data will be supplied, can be deleted. Consequently, the following lines is a valid document section, setting just one number in the formulation document:

```
TYPE;"FLEXIBLE_RECORD.Formulation"  
NAME;...;"Formulation.IdentifiedUseNumber"  
"NumericTest";...;"123"
```

When processed by the IIP, this information would

- either trigger the creation of a new instance of this document type in a selected parent entity
- or overwrite an existing document with the name "NumericTest" with the given data.



As of now, the IIP does replace existing IUCLID documents with the data found in the IEF file. There is no mode to merge / append information from the IEF file into the existing IUCLID document.

3.3.3. Field data section with one or many instances

Following the field definition line, the actual data with content of the defined fields follows.

The data for one or many instances can follow the preceding field definition for a specific document type. A data section for a new instance starts, when the mandatory first NAME column contains a value for the document name. Only the first line of a data section for one instance should contain a value in the NAME column, the remaining NAME cells of the same data section are to remain empty.

3.3.4. Field data for one document instance

This chapter explains, how the data for one given document instance (with "document" in the generic sense of IUCLID, including entities) is to be represented in the IEF:

- If a document definition only contains fields and no repeating blocks, then every field will have one cell value in the IEF file. Values for multi-picklists are concatenated with the intra-cell delimiter as described. Hence, the values of the fields for this type of document type can be written in one IEF line.
- For documents with repeating block / table rows the data can be written like in a table. The example below is a partial definition from the `FFLEXIBLE_SUMMARY.EstConcGroundwater` document type; the field names are abbreviated for better reading. The IEF representation would be very similar, with cell delimiters instead of the table borders.

UseDescription	ParentMetabolite	Substance	Tier	Model	Scenario	PecGW
GAP Use 1	parent	Cloquintocet-methyl	Tier 1	MACRO	Chateaudun	ca. 12 µg/L
GAP Use 2	metabolite	Impurity 211153	Tier 2	pearl	Hamburg	< 14 µg/L
GAP Use 3	parent	Impurity 222196	Tier 3	Pelmo	Porto	≤ 20 µg/L
GAP Use 4	metabolite	Metabolite CGA 146445 - (R,S)-2-[4-hydroxyphenoxy]propanoic acid	Tier 4	other: Tormo	Sevilla	≥ 8 µg/L
GAP Use 1	parent	Metabolite CGA 146445 - (R,S)-2-[4-hydroxyphenoxy]propanoic acid	Tier 1	MACRO	Chateaudun	ca. 12 µg/L
GAP Use 2	metabolite	Metabolite CGA 178486 - Clodinafop-propargyl (racemic mixture)	Tier 2	pearl	Hamburg	< 14 µg/L
GAP Use 3	parent	Metabolite CGA 184927 - Clodinafop-propargyl (isomer)	Tier 3	Pelmo	Porto	≤ 20 µg/L
GAP Use 4	metabolite	substance	Tier 4	other: Tormo	Sevilla	≥ 8 µg/L

The most complexity comes with nesting, when repeating blocks are nested in other repeating blocks. A good example is the OHT 85-5, which has deeply nested repeating blocks. The block definitions within the Materials and Methods section are as follows:

- Trial information (1..n)
 - Plot (1..n)
 - Application (1..n)
 - Applied Test material (1..n)

- Active ingredients (1..n)
- Sampling and residues (1..n)
 - Residue level (1..n)

Those nested blocks are handled in a similar manner as handling multiple instances: The complete data for a nested block has to be written to the IEF (in multiple lines), before any data of the next instance of the parent block is written. As an example, the first data for Plot 2 can only be written to the IEF file, when all the data for Application 1..x has been written, and recursively data for Application 2 can only be written when all the data for Applied Test material 1..y has been written; and so forth.

Parallel nestings are independent and can be written on the same IEF line. In the example above, the data in the branches Trial information and Sampling and residues are independent on another and can hence be written independently, with data of the two branches being written on the same IEF line.

The best way to learn and experience the nesting is to export this type of data in IEF format with the IIP and compare the data from IUCLID with the data in the IEF file.

3.4. End marker

An IEF file ends with the keyword **END** on a new line. This ensures that no IEF information is processed by the IIP where the generation has accidentally been interrupted.

Chapter 4. IEF syntax

The expected format of an IEF file should be aligned with [RFC4810](#)-compliant ("Common Format and MIME Type for Comma-Separated Values (CSV) Files").

4.1. File suffix

The suggested file suffix for an IEF file is ".iip". This avoids that other applications open and change the file unintentionally, like Microsoft Excel. Additionally, future client-side applications can be registered on this suffix.

4.2. Encoding

An IEF file must be encoded in UTF-8, with or without the Byte Order Mark (BOM).

4.3. Line breaks

The IEF uses and accepts CRLF (`\r\n`, Windows-style) as line separator.



This is the definition for the line breaks within the IEF file (see the example above). For line breaks **within** values please see below.

4.4. Comments

Comments can be added by starting a new line with two hashes: "##"

4.5. Cell Delimiter

The delimiter between cells in the IIP file is implicitly supplied on the first line of the IIP file. The list of supported delimiters variants is: `,` `;` `:` `=` `|`

4.6. Quoting cell values, masking of the quote character

The IIP uses the " (quotation mark) character for both quoting and masking / escaping:

- Quoting: surrounding values within cells. All cell values that may contain the cell delimiter character and / or linebreaks, need to be quoted, e.g.

```
"Indeed; this is a text  
with a CRLF line break"
```

- Escaping: To use the quoting character " inside the quoted text, it needs to be escaped using the same quotation mark character, e.g.

```
"This is a ""quoted"" text"
```

4.7. Intra-cell delimiter

The intra-cell delimiter is identical with the cell delimiter. It is used to split multiple values within one cell. Examples for multiple values in a cell are:

- Multiple "picklist" entries (with remarks)
- Multiple "docRef", "blockRef" or "attachment" entries
- "md5:" and "file:" values for one "attachment"

The intra-cell delimiter `;` needs to be escaped with `""` and the cell must be wrapped in quoted string, e.g.

```
"PreviousCell";"MyCell_V1"";MyCell_V2"";MyCell_V3";"FollowingCell"
```

4.8. Line separator in text cells

Within cell text values the line separator characters need to be represented with replacement strings:

- Carriage Return `\r`: Use `%x0D` as text value
- Line Feed `\n`: Use `%x0A` as text value

This is an exception to the RFC4810 definition; however the CSV parsing library of the IIP (openCSV) unfortunately cannot properly keep CarriageReturns within quoted content values, without creating issues with end of line content.

Chapter 5. Some IUCLID basics

For a better understanding of what the IEF does, a definition of the basic IUCLID concepts is required.



This chapter uses some textual extracts from the ECHA manuals, without explicit citation, to ease reading.

5.1. IUCLID document types

The central concept of the data representation in IUCLID is the document. The IUCLID document is a set of field-values that store scientific and/or administrative data, that conceptually belong together and represent a logical unit.

The main types of documents are:

Table 1. IUCLID document types

Type	Examples
Simple entity	Literature References, Reference Substances and Test Material
Composite entity	Mixture, Substance and Template. A composite entity can have associated section documents (often called simply "documents"). The composite entity with its associated section document is called "dataset".
Section document (often called simply "documents")	Term for the many document types that can exist within a composite entity, as OECD Harmonized Templates, Endpoint Summaries, GAP document etc.

5.2. IUCLID field types

For each IUCLID type above the set of field-values is defined as part of the [IUCLID format](#), which is updated once a year as part of major IUCLID releases. Therefore, the IEF template for a specific document type will likely change when the underlying IUCLID document definition changes.

A field definition specifies a simple data structure in the context of a document, that has a label and a field-value type. Supported field types:

Table 2. IUCLID field types

Field type	Examples
Text types	<ul style="list-style-type: none"> • text/plain (ASCII): Text (255 char.) Text (2,000 char.), Text (32,768 char.), Text template • text/html: Text (rich-text area)
Numeric types	<ul style="list-style-type: none"> • Numeric (decimal including unit) - or "quantity" • Numeric (decimal or integer) • Numeric range (decimal, precise) • Numeric range (decimal) • Numeric range (decimal with picklist) • Numeric range ("half bounded" or "half range")
List types (Picklists)	List (picklist), List sup. (picklist with remarks), List sup. (picklist with remarks - 2,000 char.), List sup. (picklist with remarks - 32,000 char.), List multi. (multi-select list), List multi. (multi-select list with remarks), List multi. (multi-select list with remarks - 2,000 char.), List multi. (multi-select list with remarks - 32,000 char.)
Link types	Link to endpoint (single), Link to endpoint (multiple), Link to entity (single), Link to entity (multiple), Link to lit. reference (multiple)
Other types	Check box (Boolean), Attachment (single), Attachment (multiple), Image upload, Date, Confidentiality, Endpoint reference representation

5.3. Field identifier

Every field in IUCLID is identified by a identifier which is unique not only in the document, but across IUCLID itself. As example, the field with the label "Endpoint" in the section "Administrative Data" in the document with the technical name `ENDPOINT_STUDY_RECORD.ResiduesInRotationalCrop` has the field identifier (or "path") `ENDPOINT_STUDY_RECORD.ResiduesInRotationalCrops.AdministrativeData.Endpoint`. The field identifiers are available in the IUCLID user interface next to each field via the info icon. Every field identifier has the technical name as prefix; then the remaining string describes the "nesting" of the field inside of the document. In the given example, the "Endpoint" field is part of the "Administrative Data" block.

5.3.1. Abbreviated field identifier

The IEF format uses an **abbreviated field identifier**, which is unique within a given document type. In the example above, the abbreviated identifier is `AdministrativeData.Endpoint` which is a

unique field identification within the document `ENDPOINT_STUDY_RECORD.ResiduesInRotationalCrop`.

5.4. Field types and display in the IUCLID web frontend

There is one single field identifier for each IUCLID field, independent on the field type. This is also true for field types, that have a complex internal structure and require the input in multiple controls, like

- full ranges, like "< 15 < 30 mg/l" (5 display controls)
- hierarchial picklists with remarks, as in "Metabolism in plants" `ENDPOINT_SUMMARY.MetabolismPlants.KeyInformation.PrimaryCrops.Commodity` (rendered in many controls dependent on the number of selected picklist items)

Internally, IUCLID processes a full range field type in a JSON structure like this:

```
{
  "lowerQualifier": "<",
  "lowerValue": 15,
  "upperQualifier": "<",
  "upperValue": 30,
  "unit": {
    "code": "113451"
  }
}
```

and a hierarchical picklist with comments as follows:

```
[
  {
    "code": "1342",
    "other": {
      "default": "Sea weed"
    },
    "remarks": {
      "default": "Test remark 1"
    }
  },
  {
    "code": "113598",
    "remarks": {
      "default": "Test remark 2"
    }
  },
  {
    "code": "113608",
    "remarks": {
      "default": "Test remark 3"
    }
  }
]
```

```
]
  }
}
```

In the IEF file definition, there is exactly one cell for each IUCLID field, independent on the internal processing of IUCLID or the display in the IUCLID user interface. The data can be provided in a user-friendly representation and the IIP takes care of the required transformation in the internal IUCLID structures and codes as shown above.

5.5. Phrasegroups

Some IUCLID fields called picklists do not accept all values, but only controlled vocabulary, which are defined in the associated phrasegroups. Each phrasegroup entry has an associated code (integer) and decode (usually textual) value. The code value is the technical internal identifier, the decode value the corresponding human-readable value.

There are a couple of variations of picklists:

Closed versus open picklists

- closed picklists, that exclusively accept values from the phrasegroup
- open picklists, that also allow to supply an arbitrary value, which is ususally prefixed by the term "other:", but eventually other prefixes. One phrasegroup may have multiple open picklist entries with different prefixes in one phrasegroup.
For an open picklist item, the prefix (usually "other" is also represented by code / decode).

Single or multiple

- Picklists can be single or multiple, accepting one or more values.

Hierarchical picklists

- Hierarchical picklist are displayed as a tree in the IUCLID interface; and only leaf nodes can be selected, not inner nodes. Internally, they are defined with a number of phrasegroups, with the decode value on one level defining the phrasegroup on the next level.

Default picklist value

- Some phrasegroups define a default value, like for many endpoint fields with picklists. The default value is set automatically when creating a document, however the value can then be deleted and the document can be saved with an empty field value.

`ENDPOINT_STUDY_RECORD.Melting.AdministrativeData.Endpoint`

In the IEF the content for picklists can be provided using both code and decode value.

5.6. Blocks and references to blocks

A block is a collection of elements (fields or blocks) defined under the path of the block. A block

itself has its own field identifier (e.g. `ENDPOINT_STUDY_RECORD.ResiduesInRotationalCrops.AdministrativeData` for the Administrative section of the respective document). Blocks can be nested, see OHT 85-5 as an example of a document definition with nesting.

In the IUCLID interface, a block is represented as a section or subsection of the document. A block can be rendered as a table; in this case each table row is a separate block nested in the main table block.

Blocks can be single or repeatable:

- Single block: A block with maximum one occurrence in the document context
- Repeatable block: A block with multiple occurrences within the same document, e.g. a table row definition

A block is a container element and does itself not carry any information that can be set by the user.

In IUCLID, a single block is uniquely identified per document via its field identifier. Repeatable blocks however have the same field identifier so they are each assigned with UUID identifiers.

There are some IUCLID documents where within a document one field references a block in the same document, e.g. the treatment information references a block from a listing of test crops in `ENDPOINT_STUDY_RECORD.MetabolismInCrops`. This is called a "blockref".

Chapter 6. Mapping of IIP values to IUCLID field types

The following chapter describe the mapping of values of different fields types done by the IIP software between values in the IEF and the IUCLID fields. This flexibility allows the applicant to upload data in a greater variety of representations as directly accepted by IUCLID.



Next to the value mapping, please ensure the required quoting and masking of values and and the treatment of line separators within values (see below). This is not mentioned here again.

6.1. Boolean

A **boolean** data field represents a simple true/false value, displayed as a checkbox in IUCLID.

Import **boolean** as accepted IEF values by IIP

One of the following options

- true/false
- t/f
- 1/0
- yes/no
- y/n

Export **boolean** from IUCLID to IEF by IIP

exported as "true"/"false"

Exemplary **boolean** data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.RobustStudy
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.UsedForClassification
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.UsedForMSDS

6.2. Date

A **date** data field represents a date, consisting of year, month and day.

Import **date** as accepted IEF values by IIP

Date format as defined by the DATE marker in the IEF header. Supported formats are:

- YYYY-MM-DD
- DD-MM-YYYY
- MM-DD-YYYY
- DD.MM.YYYY
- MM/DD/YYYY
- DD/MM/YYYY

Export date from IUCLID to IEF by IIP

exported as "YYYY-MM-DD"

Exemplary date data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.StudyPeriodStartDate
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.StudyPeriodEndDate

6.3. Numeric

A **numeric** data field represents a number value as integer or floating-point value with double precision.

Import numeric as accepted IEF values by IIP

Double: Both separators "." and "," accepted, also mixed when in different values.

Note: Also values like 1.234e5 are valid double values, "e" or "E" can be used for the exponent.

Integer: must be a valid Integer value

For some field in IUCLID, a min/max value is defined in the document definition. The supplied value is validated against this constraint.

Export numeric from IUCLID to IEF by IIP

Double: Using "." decimal separator

Exemplary numeric data fields:

- LITERATURE: GeneralInfo.ReferenceYear
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.RecoveryResultsAndCharacteristicsOfAnalyticalMethod.Recovery.NumberReplicates
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.ResultsUsingEnforcementMethod.Repeatability.MeanContent

6.4. Text

A **text** data field may contain plain or rich text, varying in length.

"Plain" text has no formatting and uses a simple input field in IUCLID.

"Rich" text is formatted textual information, displayed in rich text fields in IUCLID. Despite of the name of the field type, the technical format is not RTF, but HTML. As of now, the IUCLID format does not specify the exact version or syntax of [HTML](#). The IUCLID web frontend apparently uses the free version of the [TinyMCE](#) editor that does not apply in-depth HTML format control



The IEF does currently not apply any HTML format validation of content for rich-text fields, this has to be done by the program generating the IEF file. As ECHA plans to apply HTML validation in the future, it is advisable to apply HTML cleanup to a simple HTML syntax, prior to IUCLID upload. Please consider using a tool like [OWASP HTML Sanitizer](#) to clean your HTML prior to upload it to IUCLID.

Import **text** as accepted IEF values by IIP

Checks for max length as given in the IUCLID field type definition. For rich text fields there currently is no HTML syntax validation.

Export **text** from IUCLID to IEF by IIP

no transformation during export

Exemplary plain **text** data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.StudyPeriod
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: MaterialsAndMethods.Guideline.VersionRemarks

Exemplary rich **text** data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: MaterialsAndMethods.AnyOtherInformationOnMaterialsAndMethodsInclTables.OtherInformation
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: ApplicantSummaryAndConclusion.ExecutiveSummary

6.5. Picklist

A **picklist** data field contains one or more predefined values from a defined phrasegroup with optional remarks. a "closed" picklist only allows values from the predefined list. An "open" picklist allows custom values.

Some picklists are hierarchical and provide grouping in a tree-like structure. Within a hierarchy each picklist value has a unique code/decode, that identifies its position. There is no need to specify the path of a picklist from the root of the hierarchy.

Import picklist as accepted IEF values by IIP

Options:

- For closed picklists:
 - code
 - decode
- For phrasegroups with open picklist entries
 - apply regular expression to parse `<prefix>:<value>` syntax (prefix is usually "other:")
- For hierarchical picklists: Check decode values for leaf nodes in group containers
- Multiple: split by in-cell delimiter
- Required: Job error, if value is not present
- Remarks: split by in-cell delimiter - belongs to value before; "open" phrases have no remark input
- Validation: Check for RemarksMaxLength

picklist examples:

- decode value: `mg/l`
- decode value with remarks: `"mg/l";"remarks:custom remark"`
- open value: `other:custom/open value`
- open value with remarks: `"other:custom/open value";"remarks:custom remark"`
- multiple decode/open values with/without remarks: `"mg/l";"remarks:custom remark";"other:custom/open value"`

Export picklist from IUCLID to IEF by IIP

There are a couple of picklist variations, determining the export:

Open / closed picklist entry:

- Closed picklist entry: "Decode" value of picklist entry
- Open picklist entry: Defined prefix (usually "other:"), followed by the value.

Picklists with optional remark field: Add remark value in same cell, separated by in-cell delimiter, prefixed with string "remarks".

Multiple / hierarchical picklists: Separate entries for each picklist entry (as defined above); separated by in-cell delimiter.

Exemplary picklist data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.Endpoint
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.DataWaivingJustification
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: MaterialsAndMethods.Guideline.Guideline

6.6. Quantity



The IEF syntax of the range data field was changed in IEF version 1.1: The last unit picklist part was separated from the numeric value, the quantity now consists of two separate fields in the IEF file. Please adapt your IEF files accordingly.

A **quantity** data field contains a **numeric** with a unit **picklist** value.

Import **quantity** as accepted IEF values by IIP

- First field: regular expression for numeric value (see above)
- 2nd field: decode value of open/closed unit phrasegroup (see above).

quantity examples:

- First field **34.1 mg/l**, second field **mg/l**
- open/custom picklist unit value: First field `23`, second field **other:kcal/h**

Export **quantity** from IUCLID to IEF by IIP

- First field: numeric value (see above)
- 2nd field: decode value of open/closed unit phrasegroup

Exemplary **quantity** data fields:

- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.RecoveryResultsAndCharacteristicsOfAnalyticalMethod.Recovery.FortificationLevel
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.RecoveryResultsAndCharacteristicsOfAnalyticalMethod.Recovery.FortificationLevel

6.6.1. Picklist matching

The picklist values supplied in the IEF are matched to the IUCLID decode picklist values as follows:

- The value in IEF is compared case-sensitive and as substring match to decode picklist values. The substring matching requires at least 3 input characters.
- The value in IEF is compared case-insensitive and as substring match to the decode picklist values
- If there is exactly one match, then the IEF value is matched to that IUCLID picklist value. This

e.g. allows to use the OECD guideline numbers (e.g. "106") instead of supplying the full text as in IUCLID.

- If there is zero or more than one match, then the mapping fails and an error is raised during import.

Of course, the IEF can also contain the 1:1 same value as in IUCLID.

6.7. Range



The syntax of the range data field was changed in IEF version 1.1. If a unit is part of the IUCLID definition of the range, the unit picklist part was separated from the preceding part; then the range consists of two separate fields in the IEF file. Please adapt your IEF files accordingly.

A **range** data field represents a full or half-bounded numeric range. The full range consists of a lower qualified value, an upper qualified value and a unit **picklist** value. The half-bounded range only consists of one qualified value and a unit **picklist** value.

The bound qualifiers are optional. The unit **picklist** is only present in some **range** data fields. Please consult the definition report for details.

As the syntax differs between full and half-bounded range, the field types are described separately.

6.7.1. Half-bounded Range

Import half-bounded **quantity** as accepted IEF values by IIP

regular expression with **boundQualifier** **value** **unit**

Allowed values for **boundQualifier** are **ca.**, **<**, **≤**, **>**, **≥**

half-bounded **range** examples:

- value: **15**
- value with bound qualifier: **ca. 15** (no two fields)
- value with bound qualifier and unit: First field **ca. 15**, 2nd field **mg/l**
- value with bound qualifier and open unit: First field **ca. 15**, 2nd field **other:custom/open unit**

Export half-bounded **range** from IUCLID to IEF by IIP

- First field: **<boundQualifier> <value>**
- 2nd field: **<unit phrasegroup: decode/other>**

Exemplary half-bounded **range** data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicity:
KeyValForChemicalSafetyAssessment.AcuteToxicityViaOralRoute.EffectLevelValue
- ENDPOINT_STUDY_RECORD.AdsorptionDesorption:
KeyValForChemicalSafetyAssessment.KocAt20Celsius

6.7.2. Full Range

Import full range as accepted IEF values by IIP

A full range value has the following syntax `lowerQualifier lowerValue upperQualifier upperValue unit`. To prevent ambiguity between lower/upper value designation with incomplete full range values, the lower and upper numerical values have to be prefixed with `l` (lower) and `u` (upper) respectively.

Allowed qualifiers are typically `ca.`, `<`, `≤`, `>`, `≥`. For some full range fields the qualifiers are more restricted. Please consult the definition report for more details.

Regular expression evaluators are applied in the following order:

1. Regex with `lowerQualifier lowerValue unit`
2. Regex with `upperQualifier upperValue unit`
3. Regex with `lowerQualifier lowerValue upperQualifier upperValue unit`

full range examples:

- lower value: `l15`
- lower qualifier/value: `> l15`
- lower qualifier/value and unit: First field `> l15`, 2nd field `mg/l`
- upper value: `u30`
- upper qualifier/value: `< u30`
- lower value and upper value: `l15 u30`
- lower qualifier/value and upper qualifier/value: `> l15 < u30`
- lower qualifier/value and upper qualifier/value and unit: First field `> l15 < u30`, 2nd field `mg/l`
- lower qualifier/value and upper qualifier/value and open unit: First field `> l15 < u30`, 2nd field `other:custom/open unit`

Export full range from IUCLID to IEF by IIP

- First field: `> l25 < u30`
- 2nd field: `mg/l`

Exemplary full range data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: ResultsAndDiscussion.EffectLevels.EffectLevel
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.RecoveryResultsAndCharacteristicsOfAnalyticalMethod.Recovery.MRM MZ

6.8. Document Reference

A field of type `docRef` in IUCLID is a field that contains one or more references to other IUCLID documents. Please note that here the term "document" is used in its generic sense, meaning that everything in IUCLID is a document including substances, mixtures, references substances etc; not just the "documents" in a composite entity (dataset). Please also note the difference to the other two reference types `BlockRef` (intra-document references) and `InventoryRef` (reference to the internal IUCLID chemical inventory, used in `REFERENCE_SUBSTANCE`)

In IEF the idea is to specify the target for the reference using business values (name, cas number, ...) and not UUIDs, as these are usually not known to a source application from where the IEF information is derived from.

The IUCLID format definition, specifies which types a `docRef` is intended to reference (called `referenceTypes`). This may be

- one of the core documents, like substance, test material, reference substance etc.
- a specific type / subset of the documents in a "composite entity" (dataset). As example, an endpoint summary `ENDPOINT_SUMMARY.AcuteToxicity` for acute tox will only allow referencing endpoint study records of type `ENDPOINT_STUDY_RECORD.AcuteToxicityOral` in its field `ENDPOINT_SUMMARY.AcuteToxicityKeyValueForChemicalSafetyAssessment.AcuteToxicityViaOralRoute.LinkToRelevantStudyRecord` for the toxicity via oral route.
 - A link from / to another dataset document is only valid for the set of documents in the dataset plus the documents in the inherited template(s)

However, there are many examples for `docRef` fields that allow a set of target types. Consequently, when evaluating the value for a `docRef` in an IEF file in the connected IUCLID instance, all those potential target types need to be considered.

Although IUCLID doesn't enforce a `docRef` to only contain types of the defined `referenceTypes`, the IIP import considers strong typings. IIP rejects `docRef` values that do not match the defined `referenceTypes`.

Import `docRef` as accepted IEF values by IIP

For import, the rules vary according to the type of the target reference

- If supplied, the UUID is used for searching.
- If the target of the docref is another document (hence the reference is an intra-dataset link, eventually from/to an inherited template), then the parent identifier of the found target candidate must either be the ID of the parent entity or the ID of the inherited template(s)
- If **type** for the target entity is not supplied, all potential **referenceTypes** are searched for, according to the IUCLID format
- If **type** is supplied then it must be part of the potential **referenceTypes**
- If more than one matching target is found during job import, then the job status is set to ERROR (ERROR_IMPORT)
- Searching with empty values is supported, e.g. “name:” would be a supported search syntax

The common syntax for an accepted **docRef** value is as follows:
uuid:<uuid>;type:<type>;name:<name>

- **uuid** (optional): UUID of the referenced entity/document
- **type** (optional): IUCLID entity/document type
- **name**: Name of the entity/document to reference

There are specialized import structures for some types, aligned with the functionality provided in the IUCLID search/query API. When multiple search values are supplied, they are logically combined with AND, so the search will return entities that match to all criteria at once.

- Literature: **uuid:<uuid>;type:<type>;owner_study_no:<owner_study_no>;title:<title>**
 - **title** is mandatory, corresponds to the IUCLID field LITERATURE.GeneralInfo.Name
 - **owner_study_no** is optional, corresponds to the IUCLID field LITERATURE.GeneralInfo.CompanyOwnerStudyNo
- ReferenceSubstance:
uuid:<uuid>;type:<type>;cas_number:<cas_number>;cas_name:<cas_name>;name:<name>
 - **name** is mandatory, corresponds to the IUCLID field REFERENCE_SUBSTANCE.ReferenceSubstanceName
 - **cas_number** is optional, corresponds to the IUCLID field REFERENCE_SUBSTANCE.Inventory.CASNumber
 - **cas_name** is optional, corresponds to the IUCLID field REFERENCE_SUBSTANCE.Inventory.CASName
- Contact:
uuid:<uuid>;type:<type>;first_name:<first_name>;last_name:<last_name>;organisation:<organisation>
 - **organisation** is mandatory, corresponds to the IUCLID field CONTACT.GeneralInfo.Organisation
 - **first_name** is optional, corresponds to the IUCLID field CONTACT.GeneralInfo.FirstName

- `last_name` is mandatory, corresponds to the IUCLID field CONTACT.GeneralInfo.LastName

The usage of wildcards is supported as follows (identical to the Advanced Search masks in the IUCLID user interface):

- The search string `foo` or `foo*` will match to IUCLID values starting with `foo`
- The search string `foo` or `*foo` will match to IUCLID values containing `foo` anywhere in the value
- To enforce a full match to the complete string, enclose the search string in double quotes `"foo"`
- The search is case-insensitive, `foo` or `F00` or `Foo` will match all the possible upper or lowercase sequences of `foo`

docRef examples:

- Document by name: `type:ENDPOINT_STUDY_RECORD.AcuteToxicityOral;name:AcuteToxicityName`
- Test Material by name: `type:TEST_MATERIAL_INFORMATION;name:TestMaterialName`
- Literature by owner study number and name:
`type:LITERATURE;owner_study_no:StudyNo123;name:Acute toxicity in the rat`
- Reference Substance by CAS number and name:
`type:REFERENCE_SUBSTANCE;cas_number:CAS123;name:Water`
- Reference Substance by CAS number, CAS name and name:
`type:REFERENCE_SUBSTANCE;cas_number:CAS123;cas_name:CASName;name:Water`

Export docRef from IUCLID to IEF by IIP

Dependent on the type of the target reference the reference is exported as follows

- Default: `uuid:<uuid>;type:<type>;name:<name>`
- Literature: `uuid:<uuid>;type:<type>;owner_study_no:<owner_study_no>;title:<title>`
 - `owner_study_no`: only exported, if value available
- ReferenceSubstance: `uuid:<uuid>;type:<type>;cas_number:<cas_number>;cas_name:<cas_name>;name:<name>`
 - `cas_number` / `cas_name`: only exported, if value available
- Contact: `uuid:<uuid>;type:<type>;first_name:<first_name>;last_name:<last_name>;organisation:<organisation>`
 - `first_name`: only exported, if value available

Exemplary docRef data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.CrossReference.RelatedInformation
 - referenceTypes: ENDPOINT_STUDY_RECORD, FLEXIBLE_RECORD, FLEXIBLE_SUMMARY, ENDPOINT_SUMMARY, FIXED_RECORD
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: DataSource.Reference
 - referenceTypes: LITERATURE
- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: MaterialsAndMethods.TestMaterials.TestMaterialInformation
 - referenceTypes: TEST_MATERIAL_INFORMATION
- ENDPOINT_STUDY_RECORD.AnalyticalMethods: ResultsAndDiscussion.RecoveryResultsAndCharacteristicsOfAnalyticalMethod.Recovery.Analyte
 - referenceTypes: REFERENCE_SUBSTANCE
- SUBSTANCE: ContactPersons.ContactPerson
 - referenceTypes: CONTACT

6.9. Block

A **block** identifier is only required within IEF for the targeted fields in **blockRef** elements. The block identifiers must be valid UUIDs.

Import **block** as accepted IEF value by IIP

UUID to identify the block as a target for a **blockRef** element

Export **block** from IUCLID to IEF by IIP

UUID of the block

6.10. Block Reference

A **blockRef** is a link from one field of a document to a block within the same document (intra-document reference). The link is specified with a UUID value. It is used to identify the block to be referenced and the reference itself. It is recommended to use UUID identifiers for referencing blocks, in alignment with the IUCLID behaviour.

A valid UUID is defined in the IUCLID I6Z manual as follows: *universally unique identifier (UUID) being a 128-bit number. In its canonical textual representation, the UUID is represented as 32 hexadecimal (base 16) digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 alphanumeric characters and four hyphens).*

The IIP uses the method `java.util.UUID.fromString` to validate a UUID. The rules match the I6Z description. See <https://docs.oracle.com/javase/8/docs/api/java/util/UUID.html#toString--> for details.

Import `blockRef` as accepted IEF value by IIP

UUID of the targeted block identifier

Export `blockRef` from IUCLID to IEF by IIP

UUID of the targeted block identifier

Exemplary `blockRef` data fields:

- `ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.DataProtection`

6.11. Inventory Reference

The `inventoryRef` field type describes a reference to the internal chemical EC inventory supplied by IUCLID, for more information see <https://iuclid6.echa.europa.eu/de/inventories-iuclid> and the IUCLID user manual.

Import `inventoryRef` as accepted IEF value by IIP

search value `<q>`, in the form `<q>@<code>`. The part `@<code>` is optional, but usually "EC" to search in the EC inventory of IUCLID. A search without a code is done across all inventories (if any).

The search value `<q>` can represent any of the possible parameters: CAS Number, molecular formula, number, name

`inventoryRef` examples:

- CAS number: `56-93-9@EC`
- molecular formula: `C10H16N.Cl@EC`
- number: `200-300-3@EC`
- name: `benzyltrimethylammonium chloride@EC`

Export `inventoryRef` from IUCLID to IEF by IIP

Exported without change from IUCLID JSON with syntax `<number>@<code>`

Exemplary `inventoryRef` data fields:

There is currently just one supported `inventoryRef` data field (next to three further currently non-supported `CUSTOM_ENTITY` types): `REFERENCE_SUBSTANCE.Inventory.InventoryEntry`

6.12. Attachment

Data fields of type `attachment` are files of arbitrary type, attached at defined paths within IUCLID entities or documents. One attachment field may be defined as only a single entry or a list of entries. Attachments in IUCLID are assigned a UUID and the UUID is used as the value within the entity/document structure. IUCLID uses a MD5 hash to validate the integrity of the files.

Some **attachment** fields are restricted to only allow image file types (tiff, jfif, bmp, gif, svg, png, jpeg, svgz, jpg, webp, ico, xbm, dib, pjp, apng, tif, pjpeg, avif). Please consult the definition report to identify, which **attachment** fields use the `mimeType "image/*"`.

Import **attachment** as accepted IEF value by IIP

The MD5 hash and path to the attachment file is accepted in the following syntax:
`"md5:<md5>" ;file:<filepath>"`

- The **md5** parameter is mandatory
- The **file** parameter is optional

If a filepath is given, the MD5 hash is validated. The filepath may be a relative or absolute path to the file. The IIP runtime needs access to the filesystem the filepath points to.

If no filepath is given, the MD5 hash is used to find an existing attachments during the import, but will fail if no attachment with the given MD5 exists.

Export **attachment** from IUCLID to IEF by IIP

The IIP exports attachments files in a subdirectory "attachments". The MD5 hash and relative path are exported as the IEF value with the following syntax:
`"md5:<md5>" ;file:attachments/<filename>"`

Exemplary **attachment** data fields:

- ENDPOINT_STUDY_RECORD.AcuteToxicityDermal:
Discussion.AttachedBackgroundMaterial.AttachedDocument
- ENDPOINT_STUDY_RECORD.AnalyticalMethods:
MaterialsAndMethods.PrinciplesOfAnalyticalMethods.FlowDiagram

6.13. Data Protection

The **dataProtection** field in IUCLID consists of 3 values, with the following contents:

- confidentiality (type "picklist", PhrasegroupCode N64)
- justification (type "text")
- legislations (type "picklist", open, multiple, Phrasegroup N78)



The term **dataProtection** in crop protection submissions means something different than those fields.



The **dataProtection** fields and the respective identifiers of the picklists are currently not part of the IUCLID format definition and cannot be automatically retrieved via the IUCLID REST API (see IUCLID JIRA ticket 263293). Changes in the **dataProtection** fields in new major IUCLID releases may require an adaptation of the IEF and the IIP.

Import `dataProtection` as accepted IEF value by IIP

The handling of the three fields are as described for the specific field types `picklist` & `text` above

Export `dataProtection` from IUCLID to IEF by IIP

The handling of the three fields are as described for the specific field types `picklist` & `text` above

Exemplary `dataProtection` data fields:

- `ENDPOINT_STUDY_RECORD.AcuteToxicityDermal: AdministrativeData.DataProtection`

6.14. Field type with multiple values

The following field types can be defined in IUCLID to accept multiple values:

- Picklist
- Attachment
- Document Reference

For those fields, multiple values are put in one IEF cells, using the intra-cell delimiter - see examples above for picklists.

Next to individual fields, also blocks / table rows can be defined as multiple in IUCLID. To represent multiple values in blocks / table rows, multiple lines in the IEF file are used.

6.15. Preview: Specific mapping to avoid special characters

Some picklist values of IUCLID contain special characters, that are difficult to generate, as IUCLID requires the precise formatting. Examples are:

- the Mikrogramm character μg
 - This sign has the UTF-8 character code U+03BC, the code U+00B5 looking identical is **not** accepted by IUCLID
- square meter (m^2)
- The temperature sign next to Celsius ($^{\circ}\text{C}$) and Fahrenheit ($^{\circ}\text{F}$)
- German Umlaute or French accents (Kremsmünster, Châteaudun)

In a future release the IIP will allow to supply the information also in simplified form and will transform it into the required IUCLID representation:

- ug
- m2
- C and F
- Kremsmuenster, Chateaudun Unresolved directive in <stdin> - include::IEFImportIcons.adoc[]