IMMC Technicalities

Last Update: July 2024

Minimum requirements for the metadata set

Interinstitutional exchanges

In 2011 the IMFC subgroup agreed on the minimum set of metadata to be sent in interinstitutional exchanges. The set of transmitted metadata now supports a wide range of procedures, including legislative procedures (OLP) and non-legislative procedures (e.g. Consultation, Consent).

For a given procedure the metadata sent per the IMMC agreement must contain:

- Metadata of the procedure (id, type of procedure, legal basis)
- Metadata of the associated events (and optionally of the documents)

Any procedure (interinstitutional, internal or case law)

- contains *at least one event*,
- and the associated *events may contain documents*

Documents are modelled according to the **Common Data Mode**l – which separates information in four levels. For digital information only three items are relevant to the **first level** of the FRBR Entity-Relationship model:

- **Work**: expresses the idea, e.g. a proposal
- **Expression**: highlights characteristics of the way the work is expressed, e.g. *the proposal* is expressed in the English language
- **Manifestation**: highlights the characteristics of the way the expression for a work is expressed, e.g. the proposal expressed in the English language manifests itself in the pdf digital format.

The **second level** of the FRBR contains elements or groups which express the **custodianship** of the work, and these often relate back to **authority tables** (e.g. corporate bodies).

Through continuous improvement, the **core-metadata** (and consequently transmissions) **may** also **contain information** with regards to

- how documents or procedures relate to events (e.g. CIDOC elements such as dossiers which group documents related to events or to procedures)
- Temporal/circumstantial information of the related digital items transmitted (e.g. FRBRoo elements such as places, authors, timestamps)

For **internal procedures,** such as Initiative procedures of the European Commission,

- the *events can* be *linked to an internal procedure*
- which itself *contains at least one event*
- Which, per se, may contain associated documents

For harmonization purposes, where possible, the metadata will be subject of standardized encodings and the values provided for such elements must be taken by **authority tables** (e.g. the values of procedure types are taken from the Procedures Authority table).

Examples

For example, the **Ordinary Legislative Procedure** (OLP), involves various stakeholders and institutions which must communicate (internally) with each other to review, comment and amend documentation related to legislative proposals.

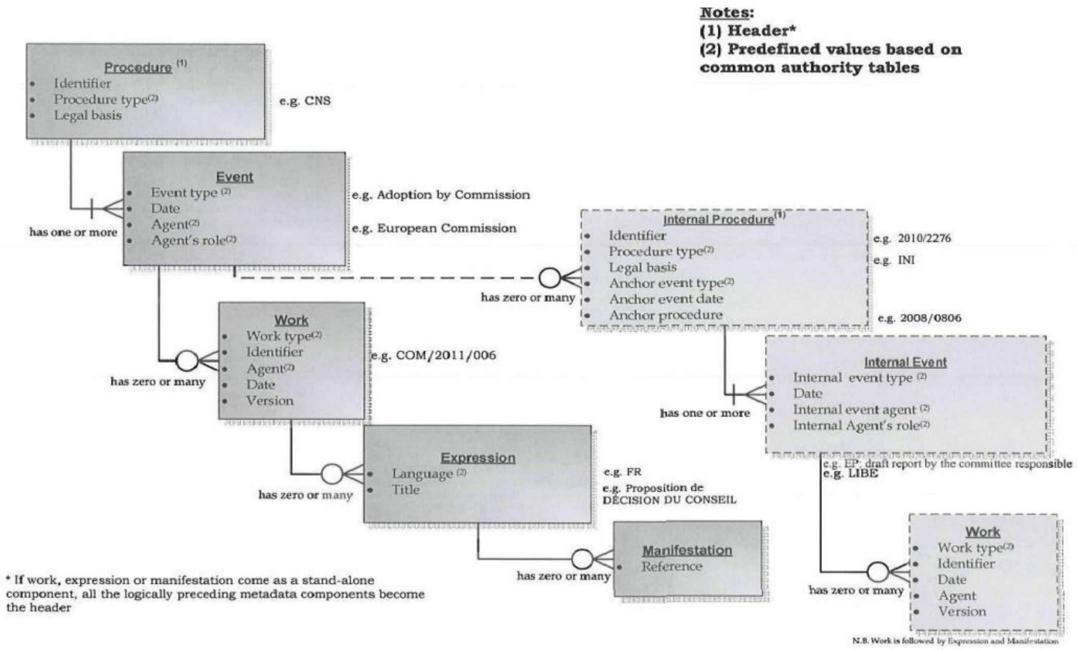
Regarding interinstitutional procedures, in the OLP

- the European Commission submits legislative proposals to the European Parliament (EP) and the Council of Ministers
- Following readings (first and the second if necessary), the European Parliament adopts a position (considering the Commission's position and the Council's comments/amendments)
- If no agreement is reached by the second reading, a conciliation committee works on a joint text.
- Once approved, the legislative act is adopted, signed and enters into force as soon as it is published on the Official Journal (OJ)

Looking into special legislative procedures such as the consultation procedure (CNS) which is used by the EU for politically sensitive issues we note that:

- The European Commission (EC) submits a proposal to the Council of Ministers and to the European Parliament.
- The EP delivers advice on the proposal (by majority of votes). They can also suggest amendments to the proposal
- The Council of Ministers can approve or disapprove the proposal based on qualified majority vote or unanimity. Although the Council of Ministers can ignore the advice given by Parliament, they must also consult the Committee of the Regions (CoR) and/or the Economic and Social Committee (EESC) if the proposal concerns a policy area relevant to these institutions.

The following picture shows a snapshot of the minimum metadata to be sent in the CNS procedure.



In

In this example, the receiver will know that:

- By means of the own-initiative procedure, the European Parliament LIBE committee has proposed a resolution and a report on an issue and has requested the EC to put forward a legislative proposal on a certain issue.
- The European Commission has adopted and submitted a legislative proposal on the specific issue, coming up with a document (proposal for COUNCIL decision) with regards to that topic (which by the work id can be deduced to be the guidelines for the employment policies of the Member States).

The sender can also append additional information to the core-metadata, to provide more context on events or works (in terms of more details or related events).

Internal exchanges

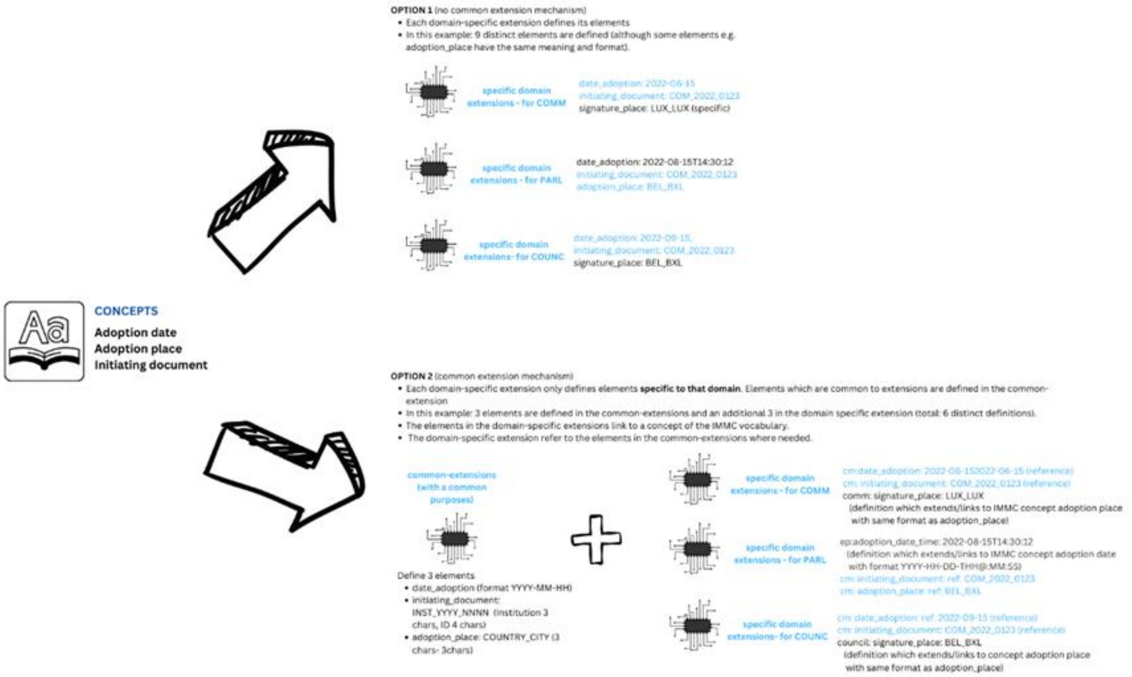
As the IMMC protocol grew in use and catered to different purposes (e.g. namely production and archiving purposes with regards to IMMCDPA), a need arose to have another set of minimum metadata (to be used on top of the original minimum set of metadata for interinstitutional exchanges). The OP Core metadata caters to the OP's internal exchanges, is available on the EU vocabularies page.

Schemas

Every IMMC schema defines its own **namespace** (mainly in URI format) and **prefix**. The prefix aids in identifying the schema the element originates from. The **core-metadata** contain information related to the FRBRoo and CIDOC elements while the transmission schema contains administrative information with regards to workflows. The **publication request schema** is a common-extension schema which can be used for messages intended for publication which do not contain any domain-specific extensions (thus merely contain core-metadata). As the institutions' needs regarding publication evolved and diverged, this file has mostly been re-implemented in the domain-specific extensions.

Overall and in compliance with information technology best-practices, the metadata (additional to the core-metadata) used exclusively for one purpose is kept locally (e.g. on domain specific extensions), while metadata (additional to the core-metadata) common to more than one domain is in the **common-extensions file**. By assembling the IMMC concepts which are common to one domain in one file, the common extension mechanism strives for consistency, as concepts which are common to multiple domains are defined once and not re-implemented locally in each extension.

Using the common-extension mechanism (Option 2) minimizes the number of times a concept has to be redefined (e.g. only distinct elements are defined). On the other hand, this approach also offers flexibility for domain-specific cases, as elements which are specific to a particular business domain can still be expressed (and are defined in the business-specific domain extension). For example, an institution such as the Commission wishing to express the place the document was signed (instead of the place the document was adopted) can still do so, by defining this element locally (in the domain-specific extension).



OPTION 1 (no common extension mechanism)

- Each domain-specific extension defines its elements
- In this example: 9 distinct elements are defined (although some elements e.g. adoption_place have the same meaning and format).

specific domain extensions - for COMM
date_adoption: 2022-06-15
initiating_document: COM_2022_0123
signature_place: LUX_LUX (specific)
specific domain extensions - for PARL
date_adoption: 2022-08-15T14:30:12
initiating_document: COM_2022_0123
adoption_place: BEL_BXL
specific domain extensions- for COUNC
date_adoption: 2022-08-15,
initiating_document: COM_2022_0123
signature_place: BEL_BXL
CONCEPTS
Adoption date
Adoption place
Initiating document

OPTION 2 (common extension mechanism)

- Each domain-specific extension only defines elements **specific to that domain**. Elements which are common to extensions are defined in the common-extension
- In this example: 3 elements are defined in the common-extensions and an additional 3 in the domain specific extension (total: 6 distinct definitions).
- The elements in the domain-specific extensions link to a concept of the IMMC vocabulary.
- The domain-specific extension refer to the elements in the common-extensions where needed.

common-extensions (with a common purposes)
Define 3 elements
date_adoption (format YYYY-MM-HH)
initiating_document: INST_YYYY_NNNN (institution 3 chars, ID 4 chars)
adoption_place: COUNTRY_CITY (3 chars- 3chars)
specific domain extensions - for COMM
cm:date_adoption: 2022-08-152022-06-15 (reference)
cm: initiating_document: COM_2022_0123 (reference)
comm: signature_place: LUX_LUX
(definition which extends/links to IMMC concept adoption place with same format as adoption_place)
specific domain extensions - for PARL
ep:adoption_date_time: 2022-08-15T14:30:12
(definition which extends/links to IMMC concept adoption date with format YYYY-HH-DD-THH@:MM:SS)
cm: initiating_document: ref: COM_2022_0123
cm: adoption_place: ref: BEL_BXL
specific domain extensions- for COUNC
cm: date_adoption: ref: 2022-09-15 (reference)
cm: initiating_document: COM_2022_0123 (reference)
council: signature_place: BEL_BXL
(definition which extends/links to concept adoption place with same format as adoption_place)

As the common extension mechanism is of an evolutive nature (i.e. the OP may re-define overlapping elements in the common-extensions), a deprecation mechanism is needed to ensure that the schemas still allow for backward-compatibility.

Technically, when elements are regrouped in the common-extension file, two additional changes must be performed on the domain-specific work level.

- In **defining** the element (element name=”date_adoption”) in the domain-specific file, the substitutionGroup element “cmext:date_adoption” is appended. Keeping “date_adoption” allows the schema to be able to validate IMMC messages which were composed using the local date_adoption element, while the “cmext:date_adoption” element allows the schema to validate IMMC messages composed with the current schema.
- Also **references** to the elements (e.g. if a file’s element is referring to the definition of another file’s element), the domain-specific namespace is replaced with the cmext namespace (as to refer to the newly defined element in the common-extensions metadata). E.g. element ref=”ecext:date_adoption” becomes “cmext:date_adoption”.

Moreover, any changes to the namespace of the element, or the name of the element, requires that the sender’s and receiver’s IT systems are adapted (to parse the message and extract the transmitted information).

The **core metadata and common extensions** pertain to metadata **Level 1**, and changes performed at this level must be agreed on by the different stakeholders in the IMFC subgroup meeting. **Domain specific extensions** instead pertain to **Level 2 metadata**, and only impact a given sender-receiver couple. Changes at this level must be agreed upon in bilateral exchanges. **Level 2 metadata** is specified in 2 schemas: the domain-specific transmission specific schema and the domain-specific extension schema. It comprises of:

- Metadata adapted from the root-level (Level 1) core schemas (core_metadata, cm_transmission and cm_common_extensions.xsd) which is redefined or extended.
- Metadata specific to the business-domain transmission.

The **authority tables** are underneath the **at** directory and all start by “at-”. Each authority table may contain the list of valid country codes, language codes, organizations among others and are managed by the OP-EU-VOCABULARIES@publications.europa.eu team.

Although the OP allows institutions to make use of the schema files in the online releases (or to download the zip package from the Downloads tab), institutions, bodies and agencies should preferably rely on the **online releases for production purposes** (available from March 2018). The desired **schema** (e.g. domain-specific transmission schema) of the online release can be found at the following address: https://publications.europa.eu/resource/distribution/{asset}/{version}/{fileformat}/schema_{assetname}/{specific-schema-path} where
- {*asset*} is immciix (formerly immc_2_core_metadata) or immcdpa (formerly immc_3_core_metadata)
- {*fileformat*} is the extension of the file (generally xsd for schemas or zip)
- {*assetname*} is immciix (former immc_v2) or immcdpa (former immc_v3)
- {*version*} follows the YYYYMMDD-N format and is mentioned in the list available under versions and
- {*specific-schema-path*} can be for example immciix-sdk-20240701-0.zip or cm_transmission.xsd.

Versioning

In line with best-practices of version-control, the following practices are to be followed:

- For each release, each schema has the same version attribute (e.g. cm-20230621-0). For authority tables, the versions are documented inside the schema (e.g. 20150715-0 for countries AT). In every case, the filename of schemas should not include the version (e.g. ep_transmission.xsd)
- IMMC messages shall refer to the oldest XSD version they can be validated by (any subsequent version will be able to validate the messages as well). This is mentioned in the cm:version_schema attribute of the IMMC descriptor.
- If the changes performed to the schema make it non compatible with backward-messages, a new IMMC schemaline is produced. The target namespace of the new schema (found at http://publications.europa.eu/resource/dataset/{asset}) should be different.
- Minor changes to the schema should not result in changes to the target namespace.

Use - Online schemas vs zip package vs individual files.

Making use of the necessary XSD (the implementation of the transmission contract) from the ONLINE links (see the IMMC Schema Compliance Criteria for URI examples) offers advantages with respect to downloading the zip package. This is because the ONLINE link refers to the *latest release* of the authority tables (via the permanent, invariant URI link to CELLAR). Consequently, the user will automatically download the schema with the link to the latest release of the authority tables, without having to perform any operations. The users can then validate the received XML descriptors against this schema.

When the user instead chooses to use download the zip package from the “Downloads” tab of the IMMC Asset page, the authority tables packaged in the zip refer to the current release. This means that when the authority tables are updated on the server-side, the user is still relying on the now-deprecated authority table files in the zip package. Users must therefore manually download the newest release of the authority-file tables before validating the structure of the descriptor. For this reason, although a user may rely on the zip package in the development phase of the project they should not rely on these schemas in the operations phase.

Please note that users should not download schemas individually for development or production purposes. These schemas are non-functional and are only made available for documentation purposes (e.g. if a user wishes to inspect schemas).

Regarding the structure of the schemas, it is important to note that:

- Domain-specific transmission (e.g. council_transmission.xsd) must be located one folder below the supporting core metadata files and on the same level as the supporting domain specific extensions. This is because the business-specific transmission file is responsible for importing all of the supporting xml files (necessary for the given exchange) and refers to these files by relative path (e.g. ../core-metadata.xsd or council_cm_extension).

In validating the XML message, the receiving system will first parse the specific transmission document which specifies the complete set of relevant XSD files to be imported. The specific transmission file is therefore always placed the first level of the folder (root) and follows the xyz_transmission_request.xsd naming convention (where xyz is the domain-specific subdirectory of the schema). Regarding this file, is important to mention that depending on the context of the exchange, not all of the XSD files are to be imported (e.g. in the context of a simple transmission of documents between the Commission and the Council, there is no need of including metadata elements from the publications.xsd file). In other words, not all XSD files within an IMMC schema release will be needed to validate a given IMMC message.

As the OP must ensure that this document must import the complete set of relevant XSD files (it it declares the XSD files to be imported, which per-se must either include all of the relevant items or must call upon and import further XSD files with the relevant elements). For example, the file may specify that the domain-specific core-metadata extension is to be imported (cm_xyz_extensions.xsd), along with the core_metadata and cm_transmission file. In turn these files may import other files. For example, the domain-specific core-metadata extension may import other files (other cm_extensions), the core_metadata file generally imports authority-table values, and the cm_transmission file may import cm_common_extensions file among others.

Also it is important to note that a domain-specific metadata extension may per-se import other domain-specific metadata extensions. Notable cases are:

- the council_cm_extension making use of some elements in the jlp_cm_extension, and therefore importing this file.
- Oj_cm_extension (and ojact_cm_extension) importing domain-extensions from the following domains: commission, council, European parliament, council of regions and eesc, European court of Auditors and generic.
- PO_cm_extension importing and making use of elements in the council_cm_extension (e.g. finding the extension with the following relative path ../council/council_cm_extensions.xsd)

Naming conventions

All **messages** exchanged within the context of the IMMC communication protocol must be wrapped in a **ZIP archive**, which **filename** must adhere to the following naming convention:

{filename}_immc.zip

where

- filename is composed of {transmission_id}_{date_time}_immc and
  - transmission_id is the unique identifier inside the IMMC transmission envelope composed by merging the IMMC transmission-id and date time attribute together.
  - date_time is the point in time of the transmission without colons and dots.

The **IMMC descriptor of the IMMC package** (e.g. sample descriptors among others) must adhere to the following naming convention:

{filename}_immc.xml

Every application that generates IMMC packages must ensure the **uniqueness of the package** filename **within the IMMC data channel.**

The IMMC protocol **recommends** a generic **naming convention for the {filename} part**

- {transmission id}_{date_time}_immc.zip
- {transmission id}_{date_time}_immc.xml

Further details are specified in the communication protocol documents.

Looking into the **directory structure:**

- Authority tables follow the yyyymmdd-0 naming convention
- In the past, IMMC schemas followed the MAJOR.MINOR(b) naming convention (with b being appended if the version is a beta version), where the MAJOR number indicates the schema-line. To date, only the European Commission's legacy public_access schemas following this convention and are frozen to v2.11.
- The other schemas naming convention do not contain the version in them, instead the file history and versioning of each schema is referenced inside the files.

With regards to releases (separate schema-lines have different release pages)

Published versions (after the 15th of April) must adhere to the following convention:
- immciix-sdk**-**<date-of-release>-<minor-version>.zip for IMMCIIX

- immcdpa-sdk-<date-of-release>-<minor-version>.zip for IMMCDPA

Published versions (before the 15th of April) must adhere to the following convention:
- cm- <date-of-release>-<minor-version>.zip for IMMC v2

- cm**3-**<date-of-release>-<minor-version>.zip for IMMC v3

Releases are handled internally, with METS packages ingested to CELLAR.

The naming conventions for the **elements** shall follow the following rules:

- A verb shall be used in order to express relationship between different entities (e.g. contains, affects, relates_to)
- A noun shall be used in order to express data properties (e.g. publisher, identifier)
- Generally, the elements follow the expression in the natural language. An exception to this rule is that the preposition “of” will be replaced by an underscore “_”. For example: date of adoption becomes “date_adoption”
- Fixed expressions will use underscores (e.g. term of office becomes “term_of_office”)
- Verbs shall be used in the imperative form for requests specified at the transmission level (e.g. associate, join, disseminate)

Groups shall adhere to the following convention: g_{group@name}
Simple (and complex) types shall adhere to the following conventions:

- t_{element@name}
- t_{attribute@name}
- t_{reusable_name}

The **specific protocol rules specification** (contract) adheres to the naming convention: IMMC<domain_name>_protocol_rules_specification.doc. The contract specifies the technical rules (rules which further complement, expand or restrict the rules specified in the schemas) and business rules (requirements by the stakeholders which express the context and the processes in the exchange). Further documentation, available on the IMMC Core Metadata Asset pages (under the Documentation Tab), expresses the parties involved for a given interaction (e.g. PlanPubli, OP, newCERES…), the workflows of the IMMC packages and standardisation requests.

Validation procedures for releases

Once a schema (or pre-schema) is published (released), the affected institutions use IMMC samples (included in the release) to validate the new schema.

When changes are performed at the Level 1 metadata, most senders/receivers must test the schema release. When changes are performed to domain-specific extensions, both the affected senders and the receiver shall test the release (this can involve multiple senders and receivers). When changes are performed at the domain-specific transmission request, just the domain-specific sender and receiver must test the schemas.

In theory, the set of sample messages is designed to be as comprehensive as possible (e.g. including messages which include *optional* elements and a diverse set of samples compliant to the contract). This would mean that all possible values and combinations of data can be tested in the Information Systems.

However, this if often not the case, as *creating the exhaustive set of instances of valid IMMC messages is often impossible (unless the transmissions' scope is quite small). This is because, from a business point of view, it is rather difficult to know a-priori all the possible messages the sender will compose. Because of this, the set of tests designed will not have complete coverage. From a technical point of view, the XSD version which is used (XSD 1.0) does not allow for exhaustive validation to occur. This is because the current XSD version lacks some features such as conditional testing and assertions, which would allow the receiver to validate some given combination of values (e.g. if element X has this value, then restriction Y and Z should apply to element F).*
To minimize disruptions at the daily operations level, the OP validates "test messages" early in the chain (upon the reception), dropping non-compliant transmissions or IMMC messages which cannot be understood (e.g. do not pass the technical validation phase).

To date, three types of tests should be performed:

- Compatibility tests of new IMMC schema release against previous IMMC messages (ensuring that existing data flows and systems sending/receiving previous schemas continue operating without restriction).
- test the new features (added to the current schema release)
- Integration tests

Regarding compatibility tests, the receivers must validate a representative set of IMMC messages (for an existing dataflow) against the IMMC schema. For validation to be considered successful, all messages must pass the validation tests.

Regarding tests which verify the validity of features, only *the bus. Stakeholders which are impacted by the new features of the schema assess whether the new IMMC instances/messages comply with the new schema (contract).* r validation to be considered successful, all messages must pass the validation tests. *Again, fo*r validation to be considered successful, all messages must pass the validation tests.

A third type of test, integration tests within the system infrastructure of each of sender and recipient(s) systems, must also be performed. However, as these can only be designed by those who have sufficient knowledge of the technical implementation of systems on the sender and recipient side, they cannot be described in this pre-liminary document. However, the OP strongly recommended that institutions test the release in system(s) of the sender and recipient.

Examples of test campaigns with regards to messages sent to OP proceed in the following way:

- the stakeholders (e.g. EP, CoR/EESC) transmit messages created by their systems during the test phase to the OP
- The OP must validate their conformity to the agreed specifications and create a report with errors/deviations from the specification documents.
- The validation report contains two parts:
  - Automatic structural validation: the IMMC instances/samples are validated against schemas. The package must be well formed, the descriptors/instances must be valid and should mention the corresponding data files (in the zip) con. If any inconsistency is found, the OP will document this in the validation report.
  - Manual business validation - In case anomalies have been flagged, an internal OP department or relevant OP stakeholder performs checks on the contents of the metadata and Word document.

The report along with the corrected set of files are then bundled in a zip and transmitted via IMMC to the sender.

Release Notes

Release notes and the respective SR (if present) are available on the IMMC Core Metadata Asset pages (under the “Release notes” tab):

- indicate standardization requests which the schema release responds to
- list the changes included in the release
- lists the impacted business stakeholders.
- lists the quality controls performed prior to delivering the release.
- Mentions the backward compatibility of the release (against previous releases).

Regarding this list we note that the following rules should be respected:

- Only stakeholders which are affected by the modifications of the schema-line should test. (e.g. if IMMCIIX changes, users of IMMCDPA do not have to test)
- If changes impact Level 1 core data, all business stakeholders shall test the samples and verify whether they are (adversely) affected by the change.
- If changes impact Level 2 (extension), only the sender and the recipient of the messages shall perform tests on the new schema with the sample messages.

It is important to note that for changes at the domain-specific extension levels the institutions which should perform the tests are the EU parties which make use of this business domain. This includes

- the EU parties which directly send or receive messages composed by this domain (e.g. for the JLP folder)
- the parties which indirectly make use of messages composed by this domain (e.g. EESC, COM..)

For changes to the domain-specific transmission schema, only the specific institution (one sender and receiver couple) which make use of this schema should test the new release.

APPENDIX

List of notable elements and values

Dossier: generic concept which groups works thematically (e.g. the Green Deal). It can link to any work (and any specific event in the context of the dossier). In IMMCIIX is included (optionally) in the core transmission metadata at the transmission level.

Session: generic concept which allows works to be grouped by meetings in which institutions decided upon them (sessions). As sessions relate to internal activities, performed to fulfil an event in the legislative procedure, they refer to works (but do not contain their manifestation). In IMMCIIX is included (optionally) in the core transmission metadata at the transmission level.

Inventory_transmission: lists all files to be transmitted along with the descriptor and also contains technical data related to the transmission (e.g. checksum, size). In IMMCIIX it is included (optionally) in the core transmission metadata at the transmission level.

Reference_procedure, Reference_dossier, reference_session: element which allows the work to reference procedures, dossiers or sessions alike. Procedures are either interinstitutional or internal, or could be court cases, dossier group works (and related events) by thematic package, sessions (e.g. parliamentary). In IMMCIIX it is included (optionally) in the core metadata at the work level.

Signatory_element: element which contains details related to who signed some document and in what context (e.g. place, role/function, date). In IMMCIIX it is included (optionally) in the core metadata at the work level.

Receipt: element in IMMCDPA which can provide feedback with regards to the transmission (e.g. received, technical validation passed...).

Service, context: elements in the core-metadata. Service mentions the unit/DG which sent the message, context relates to the IS which sent/handles the transmission.

Workflow: level in the core-metadata transmission which describes the context of the transmission (e.g. dossier’s to be mentioned) and aids in redirecting the transmission to the correct flow. For example, order or list of files to be sent (dossier) in the context of a legal procedure.

Phase_workflow: element which indicates the step in the transmission and aids in determining the action to be expected. For example: metadata (indicates the transmission only concerns an update in the metadata) and early-readings (imply that we are dealing with case-law). In IMMCDPA, this element is at the level of the core-metadata transmission while in IMMCIIX it is implemented at the domain-specific business level.

Basis_legal_procedure: part of the core-metadata and is transmitted at the event-level. Refers to the treaties which confer the necessary competencies to adopt measures at the EU level (e.g. legislative or non-legislative measures such as delegated acts). Each proposal for adoption of act must mention the article on the competences which allow the EU to act on a specified matter (e.g. Art. 114 of the TFEU, allows the EU to enact harmonization to national rules (thus to pass measures) on matters related to the functioning of the internal market by using OLP). The OLP procedure itself is described in another article (e.g. TFEU Article 294).

basis_legal_work: In the case where the work has a different legal basis to the procedure. For example, if the EP, Council or Commission consults the EESC, asking for them to issue an opinion on a matter (work) in the context of an ordinary legislative procedure (Art.114 of the TFEU), the legal basis of the opinion is Art.304 of the TFEU (which confers the above-mentioned competencies to the EESC).

Procedure_type: can be interinstitutional, internal, or case-law. The values are taken from the AT.

work_type, version: at the work level. from the resource type AT. Examples are Adoption by Commission (for work_type), and final (for version).

Event_type: legal or internal. Taken from the events AT.

Agent: of the event, internal event, or the work. Values are taken from corporate bodies or countries.

Agent role: from the role AT

Languages: at the expression level. From the languages AT.

Concept - Procedure: can contain events (and works related to events).

Concept - Interinstitutional procedure: can contain legal events

DATPRO: stands for data provisional. This value is introduced in the Authority Tables and in the schemas, to provisionally allow institutions and bodies to send messages which contain new concepts from the moment of the political decision (instead of the moment where the concept is available in the Authority Tables) whilst ensuring that the validation of the messages on the receiver's side is not disrupted. For example (introduce example). Technically, if the receiver's Information System is properly configured, the DATPRO value is designed such that once the concept is introduced in the AT tables and IMMC Vocabulary, the metadata which was sent alongside the DATPRO value can be detected, recovered and associated to the new concept.

List of Development and Production Tools

Development - XML editor program: read/perform operations on XML documents.

Development - Validating XML parser

Production - IMMCbuilder: a service, offered by the OP to institutions, which constructs well-defined and valid IMMC packages based on pre-formatted information submitted by the user. The tool can be accessed through REST services, command lines or the Web GUI. The tool fills up designated templates on the basis of the metadata filled in by the user and then bundles the files up (contents & IMMC descriptor) and sends it to the information system. Internal teams at OP also use the tool to create samples for a specific domain.

Production- IMMC2CDM rules: The incoming data and metadata ingested in CELLAR complies with the CDM (Common Data Model) ontology. This ontology, also based on the FRBR model, defines the metadata of resources published by the OP of the EU. IMMC messages must therefore be mapped into CDM (before being stored in Cellar). Although IMMC and CDM do overlap partially, in that they have common metadata elements, IMMC2CDM rules are needed to map IMMC messages to CDM. The MMC2CDM ruleset has a range of mappings (from simple 1-1 maps: IMMC element-CDM property), to more complex mappings (requiring data cleaning and the use of SPARQL to check for content in CELLAR). As IMMC and CDM are dependent, both teams are to be made aware of any development in one of the two languages. New elements in IMMC should (as much as possible) follow the naming and structure of the CDM elements, and IMMC2CDM rules should be updated according to the evolution of the two assets.

Information Systems behaviour - step-by step guide.

If you are wondering what an IS should do once it receives a message, please read on. Looking into the OP’s receiving/processing information system, CERES, we notice that it can only handle *zip packages* but sending/receiving applications such as EU-Send and TRUSTEX only handle *virtual bundles*. This means that when an EU body sends files through these applications, CERES extracts the files from the virtual bundle and packages them in a zip file. Likewise, when the OP sends files through the above-mentioned applications, any concerned package (zip file) must be converted to a bundle.

On the whole CERES must:

- Unzip the file and detect the IMMC descriptor. *Each IMMC package shall contain only one file name after the name of the package, bearing the _immc.xml extensions. If this file is not detected, the recipient shall notify the sender that an invalid package has arrived.*
- Parse the IMMC descriptor: identifying the correct implementing contract against which the message should be validated. If there are errors in extracting the elements in the contract, the recipient must report them back to the sender by requesting a redelivery
- Send an (optional) reception notification with the business identifier.
- Validate the IMMC package: following specific business automatic validation rules defined a-priori between the stakeholders. Although validation errors shall always be reported, reporting a successful validation is optional.
- Process the IMMC package: redirecting it to the correct flow (e.g. metadata flow…)
- (Optionally) respond to the recipient if necessary, providing processing status information. One or more IMMC packages (to be defined).

Please note - with regards to IMMC we talk about **bundles** when the files are sent with applications such as EU-SEND and TRUSTEX and **packages.**

Testing schemas - step-by-step guide

These steps are a prerequisite for starting with test activities:

1. Download the new schema version: a new schema version is usually announced together with a URL where the schema package can be obtained through download.
2. Unpack the schema package into a separate file system location using appropriate unzipping tools.

If the receivers merely wish to validate the IMMC messages against a schema (option 1-3) they should consider the number of messages they have to validate when planning for testing efforts. However, if they also wish to assess the robustness of the IT system, thus wishes to validate the messages against an up-and-running IT system. When planning for the time taken to perform this test, the user should also consider the complexity and the configurability of the IT system.

OPTION 1: Use an XML editor

Validate a message against the schema **using an XML editor** that provides an XML validator facility, e.g. XMLspy, Oxygen, …

1. Load the IMMC descriptor to be tested into the XML editor
2. Adapt the xsi:schemaLocation attribute to the location of the unpacked IMMC schema or use the editor's schema assignment functionality to announce the IMMC schema to be used for the validation of this message . The xsi:schemaLocation attribute on the IMMC message's root element has to contain (in that order) the namespace URI of the message's root element and the related IMMC extension transmission XSD file from the location of the unpacked IMMC schema directory, e.g. <schema-directory>/generic/gen_transmission.xsd, separated by a space character.
3. Validate using the XML editor's validation functionality
4. Assess the outcome

OPTION 2: Validate message using stand-alone tool

Validate a message against the schema using a **stand-alone validation tool**, e.g. OP's XML parser module:

1. Modify the configuration of the validation tool to point to the right schema directory and/or
2. Call the validation tool with an appropriate parametrization, indicating the IMMC message as validation subject and the IMMC schema to be used for validation
3. Assess the outcome

OPTION 3: Message validation using an online XML validator

Validate an IMMC message against the schema using an **online XML validator**. XSD files are accessible on URLs, so that they can be directly used for validation.

1. Announce the IMMC schema to the online validator in the appropriate way
2. Upload the IMMC message to be validated, indicating the schema to be used

3. Request validation
4. Assess the outcome

OPTION 4: Message validation in the concerned IT system (INTEGRATION TESTS)

Validate a message against the schema **in the concerned IT system**, usually a testing environment:

1. Use the IT system's upload or configuration mechanism to announce the new IMMC schema version
2. Handle IMMC transmission to be tested in the way the IT system usually does (this procedure should be covered by the IT system's operations manual)
3. Assess the outcome.