PUB21-0775
Source language: English / Written on: 14/09/2021
Document prepared by CIS_SPF_CAF

CAF JSON Schema

1. CAF Schema overview

Agreement information formatted based on the JSON CAF specifications must adhere to strict structural and data rules that are defined in two JSON ***schema*** files called **caf-schema.json** and **spf-lookup-data-types.json**.

- caf-schema.json — Contains the json programming definition of all data elements needed for sending or receiving a CAF or acknowledgement transaction. This schema includes all structural rules, objects (records) and data element requirements.
- spf-lookup-data-types.json — Contains definition and values of all lookup tables and common data elements that share the same data types and formatting rules.

The diagram hereafter is a shortened representation of the two CAF schemas content and their relationship. For the purpose of demonstration, a CAF transaction is comprised of multiple business entities that must adhere to specific structural and data requirements (properties section of the caf-schema.json file). The business specifications for each data element and corresponding records are defined as objects, each object being referenced by one or more business entities (e.g. "CAF Transmission Header" entity requires an object of type "file_header" (definitions section)). The separation between the business entity and the object allows us to re-use the same object specifications in multiple business entities (e.g. the unique object "territory_inclusion_exclusion" is re-used in the following business entities Territory for Term, Territory for Shares etc.).

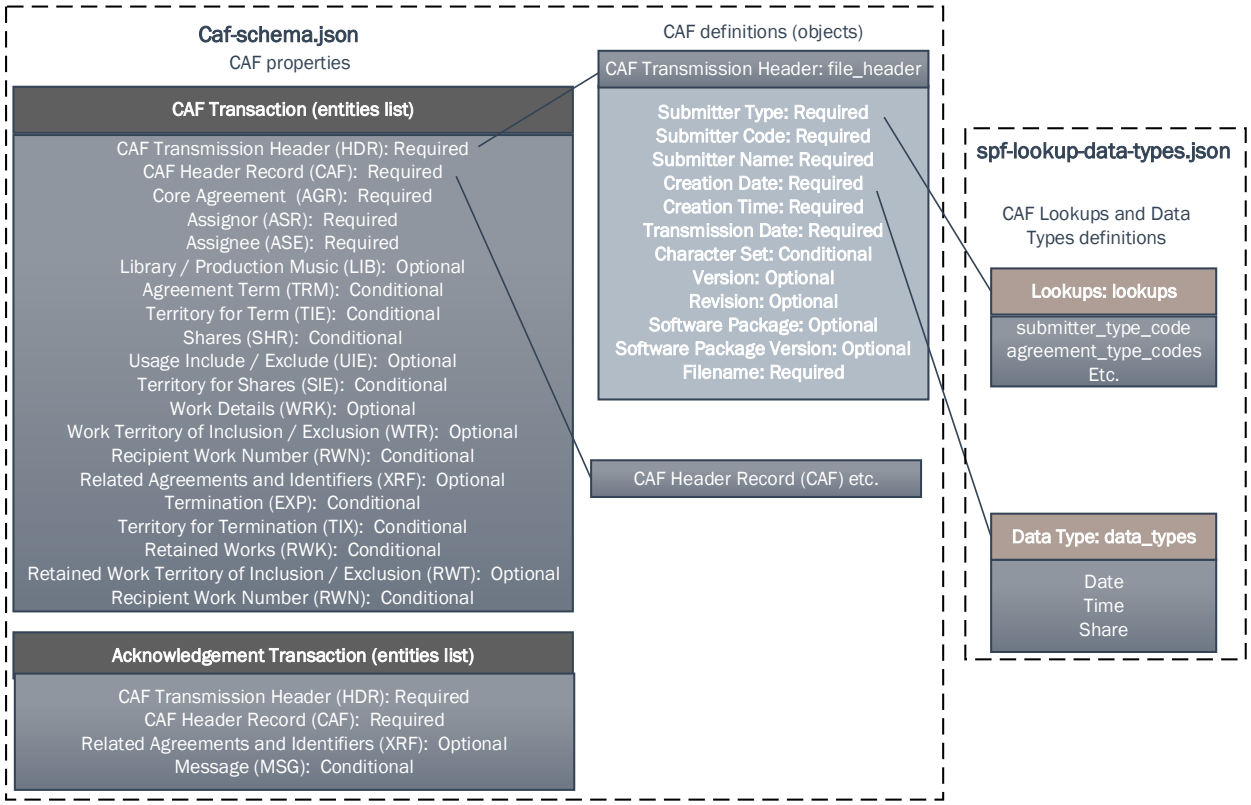


FIGURE 1: CAF SCHEMA FILES STRUCTURE AND RELATIONSHIP

Some CAF objects may include data elements that follow the same content and/or formatting rules (e.g. the Yes / No flag is a data element that is repeated in several CAF transactions -> objects). Those common data elements are defined as data_types properties in a separate schema file (spf-lookup-data-types.json). This separate schema file also includes all lookup tables required by the CAF specifications. Each lookup table is referenced at the data element level within the object definition of the schema file called caf-schema.json (e.g. file_header => Submitter Type => submitter_type_code lookup). This separation creates a technical dependency between the two CAF schemas.

2. CAF Schema network location

The CAF schemas can be found on the Society Publisher Forum website at the following web address:

https://www.societypublisherforum.net/schemas/caf-schema-3.1.5.json

This URL indicates the location of the "CAF" schema version 3.1 version revision 5.

https://www.societypublisherforum.net/schemas/spf-lookup-data-types-3.1.5.json

This URL indicates the location of the "lookup data types" schema version 3.1 version revision 5.

2.1. CAF Schema versioning and naming convention

2.1.1. Version, revision guidelines

"Release": An updated iteration of the format which is stable and deployable to users. A release may comprise a whole new version or a minor revision.

"Version" (2 blocks of digits): A new version of the CAF format signifies a substantial increase or alteration in functionality and/or structure, and likely has significant implications for senders and recipients. A new version will generally be denoted by a 0.x increment in numbering (eg; 2.4 to 2.5). In the event of a major change or complete overhaul to the format, or a succession of many small .dot changes leading to exhaustion of the .dot numbering available (last version number would logically be .9), a move up to the next whole number (ie, 2.9 to 3.0) would be warranted.

"Revision" (1 digit block): A less substantial change, not having significant impact on overall functionality or file structure. For example, changes to the text, validation rules and explanations of fields. A new revision will generally be denoted after the version number.

2.2. Schema file naming convention

CAF Schema: file name is constructed using the version and revision numbers per following construction:

Naming element	Description	Illustration
Root name	A short name describing the content of the schema.	caf-schema or spf-lookup-data-types
Version	The complete version number (including the DOT) of the CAF functional specifications used for creating the schema.	3.1
Revision	The complete revision number of the CAF functional specifications used for creating the schema.	.5
File extension	Always defaulted to json and must be preceded by a DOT.	.json

The combination of all naming elements defined in the previous table separated by the hyphen character "-" (except for version, revision and file extension) will constitute the full name of the schema file:

- caf-schema-3.1.5.json
- spf-lookup-data-types-3.1.5.json

Since the file name of the lookup tables and data types schema (spf-lookup-data-types) is physically referenced in the CAF schema file (caf-schema) as a pointer to the external lookup tables and data types definition, any change to the "spf-lookup-data-types" file name will need to be reflected accordingly in the CAF schema file (caf-schema). As a result, when considering adopting a new version or revision of the lookup tables and data types, both schemas when applied must be in updated accordingly.

3. Validation using CAF Schema

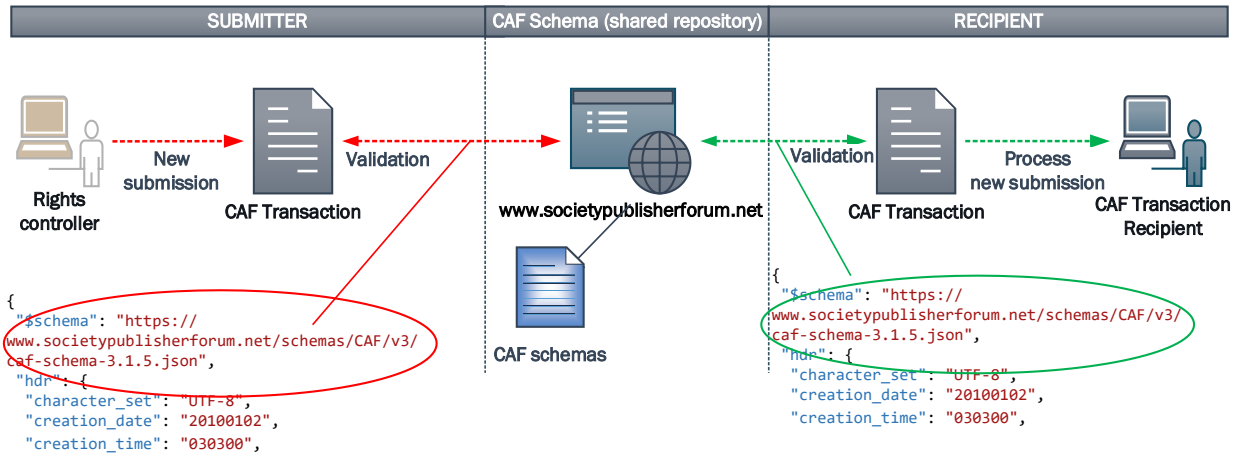
It is strongly recommended to use the CAF Schemas for validating the structure of the JSON data you are planning to send or receive. However, since JSON schema has its own limitations in terms of programming capabilities, additional semantic validations will likely need to be implemented using your organization preferred programming languages.

3.1. Declaring a JSON Schema

The $schema keyword at the top of the CAF JSON structure is used to declare the schema implemented for creating the CAF transaction. The $schema keyword must be followed by the complete CAF schema file URL provided in the previous section "CAF SCHEMAS NETWORK LOCATION". For the purpose of demonstration, it is anticipated that per diagram below:

- the submitter uses the CAF schema files available in the "***CAF Schema (shared repository)***" to validate the CAF transaction before submission to the recipient.

- upon reception of a new CAF transaction, the recipient locates the mandatory $schema property and uses the provided value to retrieve the CAF schema files from the "***CAF Schema (shared repository)***". The recipient then validates the structure of the JSON data according to the specified schema.



Note that for practical reason, the submitter may use a local copy of the CAF schema files, such practice is fine on the condition that the produced JSON CAF file include the corresponding schema file ($schema) available on the shared repository per diagram above so that the recipients can always retrieve and use the same schema files for their own structural and data validation. It is also highly recommended to always respect the schema file naming convention so that it can be found locally by the submitter and or recipient.

4. New CAF version and go-live procedure

Because the JSON CAF schema files are an integral component of the technical implementation of CAF, any changes to the CAF schemas could negatively impact the internal tools developed for the submission and reception of CAF transactions. To alleviate such risk, it is recommended to conform with the following phase approach procedure when releasing new CAF specifications.

4.1. Business requirements and approval phase

The CAF SPF working group is responsible for reviewing and approving new business requirements related to the Common Agreement Format. High level business requirements should be discussed within the CAF SPF working group and then appropriately documented for an in-depth review by the CAF SPF and CISAC working groups. Ultimately, the same working groups will formally approve the new CAF specifications. The process of reviewing and adopting the new business requirements should be done with the assistance of the technicians in charge of maintaining the CAF JSON schema.

Depending on the extend of the new business requirements, the CAF SPF working group will decide on whether a new version or a new revision of the CAF is necessary. In the same fashion, the CAF SPF working group will decide on the need and scope of the next two technical phases documented hereafter.

4.2. Technical development and validation phase

Once the business requirements formally approved, the technicians will update the CAF JSON schemas to meet the new business requirements. The change will be performed in the source control repository and the name of the updated schema file may be changed as well depending on whether a new version / revision was agreed upon by the working group. The new schema files will then be made available to the CAF SPF and CISAC working groups (when apply) to kick start the technical user acceptance test.

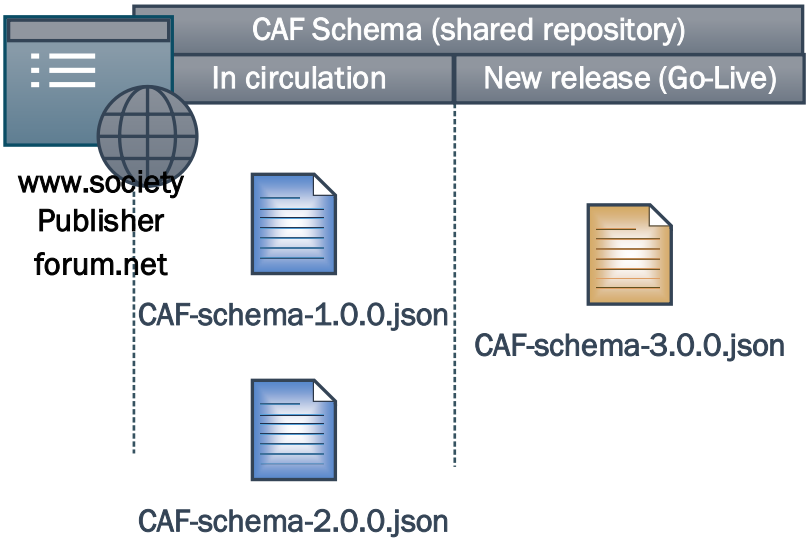
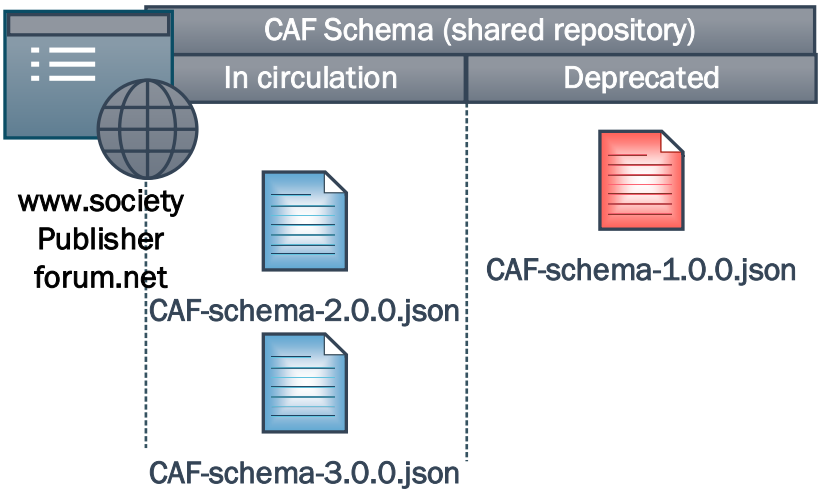
4.3. Technical user acceptance test phase

The updated schemas will be placed on the Test CAF Schema shared repository for online access by the participants of the UAT. Any technical issues will be reported to the CAF SPF working group for immediate action. Depending on the UAT result, the CAF SPF will decide on whether the updated schema is ready for Go-Live.

4.4. Go-live phase

Once the updated CAF schema(s) is ready for go-live, a formal communication is sent to the CAF SPF and CISAC working groups (when apply) to inform the various organizations about the imminent release of the new CAF Schema. The following schema activation and depreciation phases are then executed.

Schema activation: The technically vetted CAF schema(s) is then copied from the TEST shared repository into the PRODUCTION shared repository.



Schema deprecation: The oldest schema version deemed deprecated is then changed from active to deprecated status ($technicallifecyclestatus) and will remain on the shared repository for backward compatibility (Schema is deprecated, rather than immediately removed, to provide backward compatibility and to give programmers time to bring affected code into compliance with the new standard).