Citation Elements: General Concepts

26 June 2017

> *Editorial note* — This is a **first public draft** of the core part of FHISO's proposed suite of standards on Citation Elements. This document is not endorsed by the FHISO membership, and may be updated, replaced or obsoleted by other documents at any time.
>
> In particular, some examples in this draft use *citation elements* that are not even included in the draft Citation Element Vocabulary. These elements are very likely to be changed as the vocabulary progresses.
>
> The public tsc-public@fhiso.org mailing list is the preferred place for comments, discussion and other feedback on this draft.

FHISO's suite of **Citation Elements** standards provides an extensible framework and vocabulary for encoding all the data about a genealogical *source* that might reasonably be included in a *formatted citation* to that *source*.

This document defines the general concepts used in FHISO's suite of Citation Elements standards, and the basic framework and data model underpinning them. Other standards in the suite are as follows:

- **Citation Elements: Vocabulary**. This standard defines a collection of *citation elements* allowing the representation of information normally found in *formatted citations* to diverse types of source.
- **Citation Elements: Bindings for RDFa**. This standard defines a means by which *citation elements* may be identified and tagged using RDFa attributes within HTML and XML *formatted citations*, allowing a computer to extract them in a systematic manner.
- **Citation Elements: Bindings for GEDCOM X**. This standard defines extensions to the GEDCOM X data model and its JSON and XML serialisations to allow *citation elements* to be represented in GEDCOM X.
- **Citation Elements: Bindings for ELF**. This standard defines how *citation elements* should be represented in FHISO's Extensible Legacy Format (ELF), a format based on and compatible with GEDCOM 5.5, but with the addition of a new extensibility mechanism.

> *Editorial note* — Not all of these documents are yet at the stage of having a first public draft.

1 Introduction

1.1 Conventions used

Where this standard gives a specific technical meaning to a word or phrase, that word or phrase is formatted in bold text in its initial definition, and in italics when used elsewhere. The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **NOT RECOMMENDED**, **MAY** and **OPTIONAL** in this standard are to be interpreted as described in [RFC 2119].

An application is **conformant** with this standard if and only if it obeys all the requirements and prohibitions contained in this document, as indicated by use of the words MUST, MUST NOT, REQUIRED, SHALL and SHALL NOT, and the relevant parts of its normative references. Standards referencing this standard MUST NOT loosen any of the requirements and prohibitions made by this standard, nor place additional requirements or prohibitions on the constructs defined herein.

> *Note* — Derived standards are not allowed to add or remove requirements or prohibitions on the facilities defined herein so as to preserve interoperability between applications. Data generated by one *conformant* application must always be acceptable to another *conformant* application, regardless of what additional standards each may conform to.

If a *conformant* application encounters data that does not conform to this standard, it MAY issue a warning or error message, and MAY terminate processing of the document or data fragment.

Indented text in grey or coloured boxes, such as preceding paragraph, does not form a normative part of this standard, and is labelled as either an example or a note.

> *Editorial note* — Editorial notes, such as this, are used to record outstanding issues, or points where there is not yet consensus; they will be resolved and removed for the final standard. Examples and notes will be retained in the standard.

The grammar given here uses the form of EBNF notation defined in §6 of [XML], except that no significance is attached to the capitalisation of grammar symbols. *Conforming* applications MUST NOT generate data not conforming to the syntax given here, but non-conforming syntax MAY be accepted and processed by a *conforming* application in an implementation-defined manner.

1.2 Basic concepts

A **source** is any resource from which information is obtained during the genealogical research process. *Sources* come in many forms, including manuscripts, artefacts, books, films, people, recordings and websites. A full mechanism for describing *sources* is beyond the scope of this standard.

A **source derivation** is a directional link between two *sources*, incidating that the first *source* was derived from, cites or otherwise references the second *source*. The first *source* is referred to as the **derived source**, and the second the **base source**.

> *Note* — The term "derivation" is used very broadly in this standard, and includes relationships that might not normally be considered derivative. A *source derivation* exists between a digitisation, translation, transcription or index and the original document. A *source derivation* exists between a published genealogy and each *source* it cites. A *source derivation* also exists between a paper and a second paper which it is rebutting or commenting on.

A **citation** is an abstract reference to a specific *source* from which information has been used in some context. It SHOULD include sufficient detail that a third-party could readily locate the information themselves, assuming the *source* remains accessible.

A **formatted citation** is a *citation* that has been rendered into human-readable form, typically as a sentence or short paragraph that might be used as a footnote, endnote, tablenote or bibliography entry. There is no single standard on the correct form of *formatted citations*; many different style guides exist, each giving their own rules on how to construct a *formatted citation*.

> *Example* — A *formatted citation* produced for use in a footnote on the first use of the *source*, and conforming to [Chicago] might read:
>
> > [1] Christian Settipani, *Les ancêtres de Charlemagne*, 2nd ed. (Oxford: Prosopographia et Genealogica, 2015), 129–31.
>
> The [1] at the start of the citation is the hypothetical footnote number.

> *Note* — Footnotes and other reference notes sometimes contain information besides *citations*. This may include commentary on the accessibility, accuracy, authenticity or provenance of a *source*. As this information is not part of a *citation*, it is beyond the scope of this standard.

A **layered citation** is a *citation* that includes information about several *sources* between which *source derivation links* exist. The information in a *layered citation* about a specific *source*, whether the consulted *source* or one of *sources* from which it was derived, is known as a **citation layer**. A *citation* with just a single *citation layer* is called a **single-layer citation**.

The *citation layer* containing the information about the specific *source* which was consulted is known as the **head citation layer**. For a *single-layer citation*, its sole *citation layer* is necessarily the *head citation layer*.

> *Example* — A *citation* to a census return that was consulted on microfilm might contain information about the microfilm and as well as information about the census return, as in the following *formatted citation* from [Evidence Explained]:
>
> > 1810 U.S. census, York County, Maine, town of York, p. 435 (penned), line 9, Jabez Young; NARA microfilm publication M252, roll 12.

> In this example, the information before the semicolon pertains to the census return, while the information after it pertains to the microfilm. The microfilm and the census return are different *sources*, and a *source derivation* exists between them as the microfilm is derived from the census return. The information in the *citation* about microfilm forms the *head citation layer*, while the information about the census return forms a separate *citation layer*. As the *citation* contains two *citation layers*, it is an example of a *layered citation*.
>
> In this example, the *head citation layer* is not presented first in the *formatted citation*. Whether the *head citation layer* is presented first is a matter of style and emphasis, and it is common not to present the *head citation layer* first when it is a photographic or digital reproduction, as in this case.

> *Note* — *Layered citations* are often used to provide a partial statement of provenance, documenting how documents derived from one another. Many treatments of provenance also include information that is not included in citations, and hence not covered by this specification, such as a custody of ownership or characterization of the completeness of sources cited.

A **citation element** is a logically self-contained piece of information in a *citation layer* that might reasonably be included in a *formatted citation*. As this standard does not aim to provide facilities for the exhaustive description of *sources*, information about *sources* that is not normally included in *formatted citations* is not considered to be a *citation element*. *Citation elements* are represented in a sufficiently structured and language-independent way that applications can parse and reformat it in different styles and languages as needed.

> *Example* — The date that a *source* like a newspaper article was published is an example of a *citation element*. An American researcher might write the date as “Oct 8th, 2000”, while the same date might be written “zo. 8 okt. 2000” by a Dutch researcher. The *citation element* should use neither of these as its representation of the date and adopt a language-neutral format, such as one based on [ISO 8601].

The accompanying Citation Elements: Vocabulary standard defines many *citation elements*, covering the information normally found in *formatted citations* to a wide range of common *sources*. Applications MAY define their own *citation elements* or use those defined by a third-party standard; such *citation elements* are known as **extension citation elements**. *Conforming* applications MUST NOT discard unrecognised *extension citation elements*, other than at the instruction of the user, but MAY opt not to display them.

A **citation element set** is a collection of *citation elements* that completely encode the information about a *source* that is present in a particular *citation layer*.

> *Example* — The example *formatted citation* to *Les ancêtres de Charlemagne* is represented by a *citation element set* containing the following seven *citation elements*:
>
> — The author: “`Settipani, Christian`”.

> — The title: “`Les ancêtres de Charlemagne`”.
> — The edition: “`2`”.
> — The place of publication: “`Oxford`”.
> — The publisher: “`Prosopographia et Genealogica`”.
> — The year of publication: “`2015`”.
> — The page range: “`129-131`”.
>
> The footnote number is not a *citation element* as it does not pertain to the *source*. The author and page range are not expressed here in quite the same form as the *formatted citation*, but an application can readily parse them to convert them to the required format because their format is defined by this standard.

When provided with the *citation element set* for each *citation layer* in the *citation*, knowledge of which is the *head citation layer*, information about the *source derivations* between *sources* referred to in each *citation layer*, and any necessary internal state, an application ought to be able to produce algorithmically a *formatted citation* in a reasonable approximation to any mainstream citation style. If higher quality *formatted citations* are desirable, applications SHOULD allow users to manually edit them to fine-tune their presentation, and SHOULD store the result for reuse. *Formatted citations* need not include all the information from a *citation element set* if the style dictates that certain information is omitted in the relevant context.

> *Note* — Producing *formatted citations* of a professional quality following a particular style guide is a difficult art about which books have been written. This standard does not require applications to produce *formatted citations*, and throughout this suite of standards, there is no expectation that an application choosing to do so should be able to do better than a “reasonable approximation” when generating *formatted citations* automatically. That is why this standard recommends that users be allowed to fine-tune them by hand if high quality *formatted citations* are required.

Citation element sets SHOULD NOT include *citation elements* for information that is not normally included in a *formatted citation*. They are not intended to provide a general mechanism for storing arbitrary information about *sources*.

> *Example* — *Formatted citations* do not normally include details such as the email addresses, phone numbers or academic affiliations of authors, so they should not be included in the *citation element set*. A more general mechanism for describing *sources* may well include such elements, but they are beyond the scope of this standard.

1.3 Characters and strings

Characters are specified by reference to their *code point* number in [ISO 10646], without regard to any particular character encoding. In this standard, *characters* may be identified in this standard by their hexadecimal code point prefixed with “U+”.

> *Note* — The character encoding is a property of the serialisation, and not defined in this standard. Non-Unicode encodings are not precluded, so long as it is defined how characters in that encoding corresponds to Unicode characters.

Characters MUST match the `Char` production from [XML].

```
Char  ::=  [#1-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
```

> *Note* — This includes all *code points* except the null character, surrogates (which are reserved for encodings such as UTF-16 and not characters in their own right), and the invalid characters U+FFFE and U+FFFF.

A **string** is a sequence of zero or more *characters*.

> *Note* — The definition of a *string* is identical to the definition of the `string` datatype defined in [XSD Pt2], used in many XML and Semantic Web technologies.

Applications MAY convert any *string* into Unicode Normalization Form C, as defined in any version of Unicode Standard Annex #15 [UAX 15].

> *Note* — This allows applications to store *strings* internally in either Normalization Form C or Normalization Form D for ease of searching, sorting and comparison, without also retaining the original, unnormalised form.

Characters matching the `RestrictedChar` production from [XML] SHOULD NOT appear in *strings*, and applications MAY process such characters in an implementation-defined manner or reject *strings* containing them.

```
RestrictedChar  ::=  [#x1-#x8] | [#xB-#xC] | [#xE-#x1F]
                       | [#x7F-#x84] | [#x86-#x9F]
```

> *Note* — This includes all C0 and C1 control characters except tab (U+0009), line feed (U+000A), carriage return (U+000D) and next line (U+0085).

> *Example* — As *conformant* applications can process C1 control characters in an implementation-defined manner, they can opt to handle Windows-1252 quotation marks in data masquerading as Unicode. Applications MUST NOT treat non-ASCII characters as ANSEL, the character set properly used in GEDCOM, as ANSEL's non-ASCII characters do not correspond to `RestrictedChars`.

Whitespace is defined as a sequence of one or more space *characters*, carriage returns, line feeds, or tabs. It matches the production `S` from [XML].

```
S  ::=  (#x20 | #x9 | #xD | #xA)+
```

Whitespace normalisation is the process of discarding any leading or trailing *whitespace*, and replacing other *whitespace* with a single space (U+0020) *character*.

> *Note* — The definition of *whitespace normalisation* is identical to that in [XML].

In the event of a difference between the definitions of the `Char`, `RestrictedChar` and `S` productions given here and those in [XML], the definitions in the latest edition of XML 1.1 specification are definitive.

2 Citations elements

In the data model defined by this standard, a *citation element* consists of two parts, both of which are REQUIRED:

— a name, called the *citation element name*; and
— a value, called the *citation element value*.

> *Editorial note* — Earlier drafts of this standard included two other parts: a *layer identifier* and a *language tag*. The *layer identifier* has been made an implementation detail of the serialisation, and the *language tag* has been moved to the *citation element value* in the form of a *translation set*.

A *citation element set* is defined to be an ordered list of *citation elements*; *conformant* applications MAY reorder the list subject to the following constraints:

— The relative order of *citation elements* must be preserved when they have the same *ultimate super-element*.
— When a *citation element set* contains a *citation element* with the *citation element name* `http://terms.fhiso.org/sources/translatedElement`, the previous element in *citation element set* with a different *citation element name* is referred to as its **translation base**. The *translation base* of any `translatedElement` *citation element* must not change if a *citation element set* is reordered.

> *Note* — The latter requirement can be avoided by processing `translatedElements` per §3.4.1 of this standard, and then removing them from the *citation element set*.

> *Note* — Subject to these constraints, this standard allows *citation element sets* to be reordered because some serialisation languages such as JSON and RDF do not guarantee to preserve the order of elements in certain important serialisation mechanisms: for example, object members in JSON and triples in RDF other than when RDF containers are used.

2.1 Citation elements names

The **citation element name** is an identifier used to identify what information the *citation element* contains. It is a *string* that SHALL take the form of an IRI matching the `IRI` production in §2.2 of [RFC 3987].

> *Example —* The [CEV Vocabulary] defines a *citation element* for the title of a *source*. It has the *citation element name*
>
> `http://terms.fhiso.org/sources/title`

> *Note —* IRIs have been chosen in preference to URIs because it is recognised that certain culture-specific genealogical concepts may not have English names, and in such cases the human-legibility of IRIs is advantageous. URIs are a subset of IRIs, and all the *citation element names* defined by this standard are also URIs.

An IRI MUST NOT be used as a *citation element name* unless it is the *citation element name* of a *citation element* defined in the manner required by §3 of this standard.

The *citation elements* defined in this standard all have *citation element names* that begin `http://terms.fhiso.org/`. It is RECOMMENDED that any *extension citation elements* also use the `http` IRI scheme defined in §2.7.1 of [RFC 7230], and an authority component consisting of just a domain name (or subdomain) under the control of the party defining the *extension citation elements*.

> *Note —* An `http` IRI scheme is RECOMMENDED because the IRI is used to fetch a resource during *discovery*, and it is desirable that applications implementing *discovery* should only need to support a minimal number of transport protocols.

It is RECOMMENDED that an HTTP 1.1 `GET` request to a *citation element name* IRI with an `http` scheme (once converted to a URI per §3.1 of [RFC 3987]), if made without an `Accept` header, SHOULD result in a 303 “See Other” redirect to a document containing a human-readable definition of the element. It is RECOMMENDED that this definition is in HTML, and that documentation in alternative formats MAY be made available when the request includes a suitable `Accept` header, per §5.3.2 of [RFC 7231].

> *Note —* A 303 redirect is considered best practice for [Linked Data], so as to avoid confusing the *citation element name* IRI with its definition, which is found at the post-redirect URL. The *citation elements* defined in this standard are not specifically designed for use in Linked Data, but the same considerations apply.

Parties defining *extension citation elements* MAY arrange for them to support **discovery**. This when an HTTP 1.1 `GET` request to a *citation element name* IRI with an `http` scheme, made with an appropriate `Accept` header, yields 303 redirect to a machine-readable definition of the *citation element*.

> *Editorial note* — FHISO does not currently define a *discovery* mechanism, but anticipate doing so in a future standard. If such a standard is ready when this standard is released, support for *discovery* by the authors of *extension citation elements* is likely to be changed to be RECOMMENDED, but not REQUIRED, while application support for it would be OPTIONAL.

Citation element names are compared using the "simple string comparison" algorithm given in §5.3.1 of [RFC 3987]. If a *citation element name* does not compare equal to an IRI known to the application, the application MUST NOT make any assumptions on the purpose of the *citation element* or the meaning of its value based on the IRI.

> *Note* — This comparison is a simple character-by-character comparison, with no normalisation carried out on the IRIs prior to comparison. This is how XML namespace names are compared in [XML Names].

> *Example* — The following IRIs are all distinct for the purpose of the "simple string comparison" algorithm given in §5.3.1 of [RFC 3987], , even though an HTTP request to them would fetch the same resource.
>
> ```
> http://éléments.example.com/nationalité
> HTTP://ÉLÉMENTS.EXAMPLE.COM/nationalit%C3%A9
> http://xn--lments-9uab.example.com/nationalit%c3%a9
> ```

An IRI MUST NOT be used as a *citation element name* unless it can be converted to a URI using the algorithm specified in §3.1 of [RFC 3987], and back to a IRI again using the algorithm specified in §3.2 of [RFC 3987], to yield the original IRI.

> *Note* — This requirement ensures that *citation element names* can be used in a context where a URI is required, and that the original IRI can be regenerated, for example for comparison with a list of known IRIs. The vast majority of IRIs, including those in non-Latin scripts, have this property. The effect of this requirement is to prohibit the use of IRIs that are already partly converted to a URI, for example through the use of unnecessary percent or punycode encoding.

> *Example* — Of the three IRIs given in the previous example on how to compare IRIs, only the first may be used as a *citation element name*. The second and third are prohibited as a result of the unnecessary percent-encoding, and the third is additionally prohibited as a result of unnecessary punycode-encoding.

2.2 Citation elements values

The **citation element value** is the content of the *citation element*. The *citation element value* SHALL be a *translation set* if the *citation element* contains textual data that is in a particular language or script and which cannot automatically be translated or transliterated as required. Otherwise it SHALL be a *string*.

> *Example* — A book published in 2015 would have its year of publication encoded in a *citation element* with:
>
> — the *name* `http://terms.fhiso.org/sources/publicationDate`; and
> — a *value* comprising the *string* "2015".
>
> Even though an application designed for Arabic researchers might need to display the year as "٢٠١٥" using Eastern Arabic numerals, this conversion can be done entirely in the application's user interface, so a *translation set* is not required and MUST NOT be used.

2.2.1 Translation sets

A **translation set** is an ordered list of *strings*, each of which SHALL be tagged with a **language tag** to identify the language, and where appropriate the script and regional variant, in which that particular *string* is written. Each *string* in a *translation set* SHOULD contain the same information, but translated, transliterated or localised. The *language tag* SHALL match the `Language-Tag` production from [RFC 5646], and SHOULD contain a script subtags per §2.2.3 of [RFC 5646] when transliteration has occurred.

> *Example* — The `http://terms.fhiso.org/sources/title` element's value is a *translation set*. This might contain, in order:
>
> — the original title "`Η Γενεαλογία των Κομνηνών`" with *language tag* `el`, the language code for Greek in [ISO 639-1];
> — a transliteration, perhaps supplied algorithmically, with the value "`Hē Genealogia tōn Komnēnōn`" and *language tag* `el-Latn`, `Latn` being the code for the Latin script in [ISO 15924];
> — and a French translation, "`La généalogie des Comnènes`", tagged with the language code `fr`.

> *Example* — *Translation sets* are not restricted to situations where translation is not involved. They are also used where transliteration or other localisation may be needed. An author' name is rarely translated in usual sense, but may be transliterated. Andalusian historian صاعد الأندلسي might be transliterated "Ṣā'id al-Andalusī" in the Latin script. These two values would still belong in a *translation set* despite being transliterations rather than translations. They would be tagged `ar` and `ar-Latn`, meaning the Arabic language in its default script and in the Latin script, respectively. An author's names may also be respelled to con-

> form to the spelling and grammar rules of the reader's language. An Englishman named Richard may be rendered "Rikardo" in Esperanto: the change of the "c" to a "k" being to conform to Esperanto orthography, while the final "o" marks it as a noun. The respelling would be tagged `eo`, the language code for Esperanto.

> *Note* — Frequently *translation sets* will contain only a single *string*, and often most of the *strings* in *translation sets* in a given document will be in the same language.

Although the *language tags* is REQUIRED, it need not be explicit in the serialisation. A serialisation format MAY provide a mechanism for stating the document's default *language tag*, and MAY provide a global default which SHOULD be a language-neutral choice such as `und`, defined in [ISO 639-2] to mean an undetermined language. In the absence of an explicit or implicit *language tag*, applications MUST NOT apply their own default, and MUST treat the *string* as if it had the *language tag* `und`.

> *Example* — The [CEV RDFa] standard provides a means for *citation elements* to be extracted from HTML, and uses HTML's `lang` attribute to provide a default *language tag* for the document or a part of the document. Thus, if the document begins `<html lang="pt_BR">`, it is not necessary to tag each *string* separately for them to be understood to be in Brazilian Portuguese. HTML does not define a default *language tag* that applies in the absence of a `lang` tag, and applications MUST NOT apply one.

Where possible, the first *string* in the *translation set* SHOULD be the untranslated, and ideally untransliterated form of the *citation element value*. If it is known that the only available values are translations, the first *string* in the *translation set* SHOULD be an empty string tagged with the *language tag* `und`, and the translations listed afterwards. An empty *string* in a *translation set* means that its value is unknown, rather than that this particular translation is literally an empty string.

Conformant applications MAY reorder the *translation set*, but MUST leave the first *string* first, so that applications wishing to use the original, untranslated, untransliterated form can do so.

> *Note* — A standard MAY define a serialisation format that does not preserve the order of a *translation set*, but MUST take alternative steps to record the original version. For example, the language map in [JSON-LD] is very similar to a *translation map*, except that JSON's object notion, as given in §4 of [RFC 7159], does not preserve order. One possible solution is to append some private use subtag (per §2.2.7 of [RFC 5646]) to the first *language tag*.

A *translation set* MUST NOT contain more than one *string* with the same *language tag*. If an application encounters a *translation set* with duplicate *language tags*, it SHOULD prefer the first non-empty *string* with that *language tag*, and MAY *deduplicate* the *translation set*; where possible it SHOULD NOT *deduplicate* a *translation set* that has been reordered from its serialised form.

To **deduplicate** a *translation set*, the application SHALL discard all *strings* other than the first non-empty *string* with any given *language tag*. Before discarding any *strings* the application SHALL note the *language tag* of the first *string* in the *translation set*. If a *string* with that *language tag* remains after *deduplication*, the application SHALL ensure it is the first *string* in the *deduplicated translation*

set; if there is not, the application shall insert any empty *string* with that *language tag* as the first *string* in the *translation set*.

If an application needs to **merge** two or more *translation sets*, the contents of each *translation sets* SHALL be combined in order, and the application MUST *deduplicate* the resultant *translation set*.

> *Editorial note* — An earlier draft of this standard put the *language tag* in the *citation element*, and made the *citation element value* a list. This had the problem that all list values had to be available in all languages or scripts. This caused problems with lists of authors containing names in different native scripts.
>
> The earlier draft also said that the original untransliterated, untranslated value should not have a *language tag*. This allowed applications to pick out the original version, but left no way of distinguishing between translated and transliterated versions.
>
> These problems are solved in this version.

If *translation sets* are being serialised in XML, it is RECOMMENDED that the special `xml:lang` attribute defined in §2.12 of [XML] is used to encode the *language tag*.

Applications SHOULD apply *whitespace-normalisation* to any *string* in a *citation element value*. This applies both to *strings* in *translation sets* and when they are the *citation element value* directly.

3 Defining citation elements

In addition to describing the intended purpose of the *citation element*, the definition of a *citation element* (regardless of whether it is one of those defined in this standard, or whether it is a *conformant extension citation element*) SHALL state:

- its *citation element name* (an IRI);
- whether it is a *sub-element* of some other *citation element*, and if so which one;
- its *range*: the formal *class name* of its value space;
- its *cardinality*: that is, whether it is *single-valued* or *multi-valued*; and
- its *translatability*: whether its *value* is a *translation set*.

3.1 Sub-elements

A *citation element* MAY be defined as a **sub-element** of another *citation element*, referred to as its **super-element**. This is used to provide a refinement of a general *citation element*. If an application is unfamiliar with the *sub-element* it MAY process it as if it were the *super-element*, with its *value* unchanged. The *sub-element* must be defined in such as way that this only results in some loss of meaning, and does not imply anything false about the cited *source*.

> *Example —* The [CEV Vocabulary] defines a *citation element* with the name
>
> `http://terms.fhiso.org/sources/creatorName`
>
> which contains name of a person, organisation or other entity who created or contributed to the creation of the *source*. Several *sub-elements* of it are defined, including
>
> `http://terms.fhiso.org/sources/interviewerName`
>
> which contains the name of an interviewer when the *source* is an interview. An interviewer can certainly be considered to have contributed to the creation of the interview.
>
> The [CEV Vocabulary] also defines a *citation element* with the name
>
> `http://terms.fhiso.org/sources/recipientName`
>
> which contains the party to whom a *source* such as a letter is addressed. In many respects it is similar to the *sub-elements* of `creatorName`, but because a recipient of a letter cannot be said to have created or contributed to the creation of the letter, and might not even be aware of its existence if it were not delivered, the `recipientName` element cannot be defined as a *sub-element* of `creatorName`.

The *range* and *translatability* of a *sub-element* SHALL be the same as that of its *super-element*.

> *Editorial note —* The *range* of a *sub-element* could be allowed to be a sub-class of the *super-element's range*, where a sub-class is understood to reduce the value space of the *class*. (It would correspond to concept of a restriction in §2.4.3 of [XSD Pt2].) At the moment there is no clear use case for this.

Any *sub-element* of a *single-valued super-element* MUST be *single-valued*.

A *citation element's* **super-element list** is an ordered list of IRIs defined inductively as follows. If the *citation element* is not a *sub-element*, then its *super-element list* contains just its *citation element name*. Otherwise, its *super-element list* is the *super-element list* of its *super-element* to which its own *citation element name* is appended.

A *citation element's* **ultimate super-element** is defined as the first IRI in its *super-element list*.

> *Note* — This definition is equivalent to following the (possibly empty) chain of *super-elements* until it reaches something that is not a *sub-element*. It is used in specifying how applications are permitted to reorder *citation element sets*.

The **ultimate single-valued super-element** of a *single-valued citation element* is defined as the first IRI in its *super-element list* that is the name of an *citation element* that is *single-valued*.

> *Note* — This definition is equivalent to following the (possibly empty) chain of *super-elements*, stopping at the last *single-valued* element in the chain. It is used in specifying the constraints on *sub-elements* that are *single-valued*.

The **most-refined common super-element** of a collection of *citation elements* is defined as the last IRI that appears in the *super-element list* of every *citation elements* in the collection. It is only defined for *citation elements* that share a *ultimate super-element*.

> *Note* — This definition is equivalent to following the chains of *super-elements* for each *citation element*, stopping at the first element that appears in each chain. It is used in specifying how to *merge citation elements*.

3.2 Range

The **range** of a *citation element* is a **class**, which is a formal description of the set of possible *citation element values* for the *citation element*, giving both their lexical form and their semantics. *Classes* are identified by a **class name** which SHALL take the form of an IRI.

> *Example* — The [CEV Vocabulary] defines a *class* for representing the names of authors and other people, which has the *class name*
>
> `http://terms.fhiso.org/sources/AgentName`
>
> It is the *range* of several *citation elements* including
>
> `http://terms.fhiso.org/sources/editorName`

> *Note* — This definition of a *class* is sufficiently aligned with the XML Schema's notion of a simple type, as defined in [XSD Pt2], that they MAY be used as the *range* of *citation elements*. Best practice on how to get an IRI for use as the *class name* of XML Schema types can be found in [SWBP XSD DT].

The *class name* for the *class* of *strings* is:

`http://www.w3.org/2001/XMLSchema#string`

If an application encounters a *citation element value* that does not conform to the definition of the *class* used as the *range* of the *citation element*, it MAY remove the *citation element* or MAY convert it to a valid value in an implementation-defined manner.

> *Example* — The *range* of the `http://terms.fhiso.org/sources/publicationDate` element defined in the [CEV Vocabulary] is an [ISO 8601]-compatible date. An application encountering a date "8 Okt 2000" in a `publicationDate` element in dataset that uses German as its default language MAY convert this to "`2000-10-08`".

3.3 Cardinality

The **cardinality** of a *citation element* records how many semantically distinct values it can have. A **multi-valued** *citation element* is one that can logically have multiple values in a single *citation*. It SHOULD be reserved for situations where the values genuinely contains different information, and not used to accommodate transliterations, translations, or variant forms of something that is logically a single value. *Citation elements* that are not *multi-valued* are **single-valued**.

> *Example* — The `http://terms.fhiso.org/sources/title` *citation element* is defined to be *single-valued*, as *citations* do not refer to the same *sources* by multiple titles (though they may translate or transliterate the title), so a *citation element set* MUST NOT contain more than one `title`; but it MAY contain several `http://terms.fhiso.org/sources/authorName` *citation elements*, as that is defined to be *multi-valued* to accommodate *sources* with several authors.

Multiple instances of *single-valued citation element* in the same *citation element set* with the same *ultimate single-valued super-element* are known as **duplicate citation elements**. *Citation element sets* SHOULD NOT contain *duplicate citation elements*, and an application MUST NOT knowingly create *duplicate citation elements*. When *duplication citation elements* are present, they can be **deduplicated** according to the rules below.

> *Note* — Applications might inadvertently create *duplicate citation elements* when they do not know the *super-element* or *cardinality* of *extension citation elements*.

If an application encounters a *duplicate citation element* that is known to be not *translatable*, the application SHOULD favour the first of the *duplicate citation elements* and MAY *deduplicate* the *citation element set* by discarding the other *duplicate citation elements*.

If an application encounters a *duplicate citation element* that is either known to be *translatable* or whose *translatability* is unknown, the application SHOULD *deduplicate* the *citation element set* by replacing the *duplicate citation elements* with a single replacement *citation element* with the following properties:

— a *citation element name* which SHALL be the *most-refined common super-element* of the *duplication citation elements*; and
— a *citation element value* which SHALL be a *translation set* created by *merging* the *translation sets* of each *duplicate citation element*.

> *Note* — There is no requirement for an application to check for *duplicate citation elements* and *deduplicate* them other than when merging *citation element sets*, though an application MAY do so at other times. In particular, it might be advisable for an application to do this when importing third-party data, or if it has recently learnt of new *extension citation elements*.

> *Editorial note* — This standard needs to define how to merge *citation element sets*. The following text is a start towards that.
>
> > If an application needs to **merge** two or more *citation element sets*, the contents of each *citation element set* shall be combined in order. The application SHALL identify any sets of *duplicate citation elements* in the combined *citation element set* and *deduplicate* them according to the rules above. An application MAY use one or more *discovery* mechanism to attempt to obtain machine-readable definitions of any *extension citation element* used in the *citation element set* before identifying *duplicate citation elements*.
>
> However the merger of *multi-valued* elements requires thought too. Even though the data model doesn't require deduplication, it is still necessary to prevent duplication of, say, authors.

3.4 Translatability

If a *citation element* is defined to be **translatable**, then its *citation element value* SHALL be a *translation set*, and the *citation element's range* applies to each *string* in the *translation set*. If it is not *translatable*, then the *citation element value* SHALL be a single *string*. *Citation elements* with non-textual *citation element values* such as numbers or dates MUST be defined as not *translatable*.

If an application encounters a *citation element* which is known to be not *translatable*, but whose *citation element value* is a *translation set*, the application MAY convert the *translation set* to a *string* by discarding all but the first *string* in the *translation set*. If the *translation set* contains only one *string*, and if that *string* conforms to the *range* of the *citation element*, this conversion SHOULD be done.

> *Note* — This situation may arise when an *extension citation element* has been serialised in a *list-flattening format* by an application that does not know whether it is *translatable*, and subsequently read by an application that knows it not to be *translatable*.

If an application encounters a *citation element* whose *citation element value* is a *string*, but where the application knows the *citation element* to be defined as *translatable*, the application SHOULD convert the *string* to a *translation set* by tagging it with the *language tag* und (defined in [ISO 639-2] as representing an undetermined language).

> *Editorial note* — This scenario should not arise when data has consistently been processed by *conformant* applications.

3.4.1 List-flattening formats

Conformant applications MUST support *citation elements* that are both *multi-valued* and *translatable*, and MUST ensure that the *translation set* for each *citation element value* remains separate.

> *Example* — The `authorName` *citation element* is defined to be both *multi-valued* and *translatable* because a source may have multiple authors, each of whom may have names that have been transliterated into different scripts. Suppose a researcher wants to cite the Anglo-Japanese Treaty document of 1902 which was (at least nominally) authored by the Marquess of Lansdowne and Count Hayashi Tadasu whose name is written in kanji as 林 董.
>
> The following JSON serialisation is not allowed as it flattens *translation set* so it is no longer possible to determine how many authors there are, and which names are translations of which others.
>
> ```
> [ { "name": "http://terms.fhiso.org/terms/title",
>     "lang": "en",      "value": "The Anglo-Japanese Treaty" },
>   { "name": "http://terms.fhiso.org/terms/authorName",
>     "lang": "en",      "value": "Lord Lansdowne" },
>   { "name": "http://terms.fhiso.org/terms/authorName",
>     "lang": "jp",      "value": "林 董" },
>   { "name": "http://terms.fhiso.org/terms/authorName",
>     "lang": "jp-Latn", "value": "Hayashi Tadasu" } ]
> ```
>
> This is an example of a *list-flattening format* that does not conform to this specification; a *list-flattening format* that does conform to this specification is found in the next example.

A serialisation format that does not keep the *translation sets* of each *citation element value* separate is called a **list-flattening format**, and this standard provides a facility to allow such formats to comply with this standard by introducing a special *citation element* with the following properties:

Name	`http://terms.fhiso.org/sources/translatedElement`
Range	`http://www.w3.org/2001/XMLSchema#string`
Cardinality	multi-valued
Translatability	translatable
Super-element	*none*

In a *list-flattening format*, an application MUST consider every value to be a separate *citation element value*, and therefore to be a *translation set* with one element.

> *Note* — In most cases this assumption is expected to be valid. *Citation element sets* are expected to include translated or transliterated elements less often than not.

When a *translation set* with two or more *strings* needs to be serialised in a *list-flattening format*, the first *string* MUST be serialised according to the normal rules of the format, and subsequent *strings* MUST be serialised as if they were separate *citation element*, but with the `translatedElement` *citation element name* in place of the actual *citation element name*. This special *citation element* indicates that its value is not a distinct *citation element* and SHOULD instead be appended to the *translation set* of its *translation base* (i.e. the last preceding *citation element* which is not a `translatedElement`), and the `translatedElement` removed from the *citation element set*.

Example — The hypothetical JSON serialisation in the last example can be fixed by using a `translatedElement` to serialise the transliterated version of Hayashi's name:

```
[ { "name": "http://terms.fhiso.org/terms/title",
    "lang": "en",      "value": "The Anglo-Japanese Treaty" },
  { "name": "http://terms.fhiso.org/terms/authorName",
    "lang": "en",      "value": "Lord Lansdowne" },
  { "name": "http://terms.fhiso.org/terms/authorName",
    "lang": "jp",      "value": "林 董" },
  { "name": "http://terms.fhiso.org/terms/translatedElement",
    "lang": "jp-Latn", "value": "Hayashi Tadasu" } ]
```

The two `authorName` element are assumed to be separate *citation elements* and therefore to refer to different authors. The use of `translatedElement` signifies that this is not a different author. It immediately follows an `authorName` *citation element* with the value 林 董, and its value ("Hayashi Tadasu", tagged as `jp-Latn`) should be appended to that *translation set*.

Note — This standard does not say when the processing of `translatedElements` occurs. Ideally an application SHOULD do it during the process of reading a *list-flattening format*, but MAY do it later or not at all. If the application subsequently serialise the data in a non-*list-flattening format*, the `translatedElements` MAY still be present. Therefore applications reading non-*list-flattening format* SHOULD cope with the possibility of `translatedElements` being present.

If the *translation base* does not have a *translation set* as its *citation element value* (i.e. if its value is just a *string*), the `translatedElement` SHOULD be ignored and MAY be removed from the *citation element set*. If the *translation base* is a *translation set* that already contains a string with the same *language tag*, an application MUST NOT overwrite or duplicate a *language tag*; the `translatedElement` SHOULD be ignored and MAY be removed from the the *citation element set*.

The use of *list-flattening formats* is NOT RECOMMENDED except where there is a good technical reason. The use of `translatedElements` other than in *list-flattening formats* is NOT RECOMMENDED.

4 Layered citations

In the data model defined in this standard, a *citation layer* a *citation layer* is represented with two components, both of which MUST be present:

— a **layer identifier** to allow the *citation layer* to be referenced within this data model; and
— a *citation element set* containing the information in the *citation layer*.

A *citation* is represented with the following three parts:

— an ordered list of one or more *citation layers* encoded as above;

— the *layer identifier* of the *head citation layer*; and
— an unordered set of *layer deriviation links* encoding the *source derivations* between *sources* represented by the *citation layers*.

The *layer identifier* of each *citation layer* SHALL be unique within a given *citation*. It exists only to provide a means of referring to *citation layers* in *layer derivation links* and when identifiying the *head citation layer*; its value MUST NOT be used in other contexts. Applications MAY re-assign *layer identifiers* at any time.

> *Note* — This standard places no restriction on the form of a *layer identifier*. Implementations may use integers, IRIs or other convenient *strings*, but they may also use other means such as pointers to data structures in memory to represent the links represented in this standard by *layer identifiers*. Serialisation formats will place their own restrictions on the form of a *layer identifier* which may different between serialisation formats.
>
> In the common case of a *singe-layer citation*, the set of *layer derivation links* will be empty. In this case, the *layer identifier* of the *citation layer* is immaterial and an empty string could be used. This means that a *single-layer citation* can be represented using just a *citation element set*.

Applications SHOULD NOT reorder the list of *citation layers*, other than at the request of the user. The order of the *citation layers* is an indiciation of the preferred order for displaying the *citation layers*, and SHOULD begin with the one considered most important which need not necessarily be the *head citation layer*. Applications MAY ignore this order when displaying or formatting *citation layers*.

> *Note* — This is not an absolute prohibition on reording, and *conformant* applications MAY use a technology that does not preserve the order of the *citation layers*.

4.1 Layer derivation links

When the *sources* represented by two *citation layers* are linked by a *source derivation*, a **layer derivation link** is used to encode this. It has three parts, all of which are REQUIRED:

— the *layer identifier* of the *citation layer* representing the *derived source*; the
— the *layer identifier* of the *citation layer* representing the *base source*; and
— the *source derivation type*, which is an IRI used to describe the nature of the *source derivation*.

The two *layer identifiers* in the *layer derivation link* SHALL refer to *citation layers* present in the current *citation*. If an unknown *layer identifier* is present, applications MAY discard the *layer derivation link*.

The **source derivation type** SHALL be either an IRI defined in accordance with a future FHISO standard on source derivation types, or the following IRI which represents the most general case of derivation supported in this data model:

```
http://terms.fhiso.org/sources/derivedFrom
```

Applications MAY discard any IRI that it knows does not conform to the above requirement.

> *Editorial note —* FHISO intend to produce a Source Derivation Vocabulary standard giving a standard vocabulary of source derivation terms, for things like transcription, abstraction, translation, indexing, referencing, analysing, commenting on and rebutting. These will be sub-types of the `derivedFrom` *source derivation type*. The Source Derivation Vocabulary standard will also provide a mechanism for third parties to provider their own **extension source derivation types**, and provide a means of determining whether a given IRI is a *source derivation type*. If this document is ready for standardisation at the same time as this document, the previous paragraph will be updated to reference it.

4.1.1 Requirements for layer derivation links

> *Note* — The represention of a *citation* in this data model is equivalent to a directed graph whose vertex set is the set of *citation layers*, and whose edge set is the set of *layer derivation links*. Each edge is labelled with its *source derivation type*, while one vertex is labelled as the *head citation layer*. This graph is called the **citation layer graph**.

A *citation layer* is **directly derived** from another *citation layer* if there exists a *layer derivation link* whose first *layer identifier* is that of the former *citation layer* and whose second *layer identifier* is that of the latter *citation layer*. The **direct base citation layer set** of a *citation layer* is the set of *citation layers* from which the first *citation layer* is *directly derived*.

The **complete base citation layer set** of a *citation layer* is defined recursively as follows. The *citation layer* itself is part of its *complete base citation layer set*. It also contains every *citation layer* in the *complete base citation layer set* of every *citation layer* in its *direct base citation layer set*.

> *Note* — This definition is simply makes the *complete base citation layer set* the transitive closure of the *direct base citation layer set*. It contains the *citation layer* itself together with every *citation layer* from which it is derived, directly or indirectly.

The *complete base citation layer set* of the *head citation layer* SHALL contain every *citation layer* in the *citation*. If an application encounters a *citation* for which this is not the case, it MAY discard any *citation layers* that are not in the *complete base citation layer set* of the *head citation layer*.

> *Note* — This requirement says that the *head citation layer* must be derived, directly or indirectly, from every other *citation layer* in the *citation*. There MUST NOT be additional *citation layers* that are unconnected to the *head citation layer*, or which are only derived from it. In graph theory terms, this is equivalent to saying the *citation layer graph* MUST be connected, and that every *citation layer* must be reachable from the *head citation layer*. This standard does not prohibit there being additional *layer derivation links* besides those needed to ensure these conditions, and in particular does not require that the graph be acyclic.

5 References

5.1 Normative references

[ISO 10646]

ISO (International Organization for Standardization). *ISO/IEC 10646:2014. Information technology — Universal Coded Character Set (UCS).* 2014.

[ISO 15924]

ISO (International Organization for Standardization). *ISO 15924:2004. Codes for the representation of names of scripts.* 2004.

[ISO 639-1]

ISO (International Organization for Standardization). *ISO 639-1:2002. Codes for the representation of names of languages — Part 1: Alpha-2 code.* 2002.

[ISO 639-2]

ISO (International Organization for Standardization). *ISO 639-2:1998. Codes for the representation of names of languages — Part 2: Alpha-3 code.* 1998. (See http://www.loc.gov/standards/iso639-2/.)

[RFC 2119]

IETF (Internet Engineering Task Force). *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels.* Scott Bradner, 1997. (See http://tools.ietf.org/html/rfc2119.)

[RFC 3987]

IETF (Internet Engineering Task Force). *RFC 3987: Internationalized Resource Identifiers (IRIs).* Martin Duerst and Michel Suignard, 2005. (See http://tools.ietf.org/html/rfc3987.)

[RFC 5646]

IETF (Internet Engineering Task Force). *RFC 5646: Tags for Identifying Languages.* Addison Phillips and Mark Davis, eds., 2009. (See http://tools.ietf.org/html/rfc5646.)

[RFC 7230]

IETF (Internet Engineering Task Force). *RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing.* Roy Fieldind and Julian Reschke, eds., 2014. (See http://tools.ietf.org/html/rfc7230.)

[RFC 7231]

IETF (Internet Engineering Task Force). *RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content.* Roy Fieldind and Julian Reschke, eds., 2014. (See http://tools.ietf.org/html/rfc7231.)

[UAX 15]

The Unicode Consortium. "Unicode Standard Annex 15: Unicode Normalization Forms" in *The Unicode Standard, Version 8.0.0.* Mark Davis and Ken Whistler, eds., 2015. (See http://unicode.org/reports/tr15/.)

[XML]

W3C (World Wide Web Consortium). *Extensible Markup Language (XML) 1.1*, 2nd edition. Tim

Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, and John Cowan eds., 2006. W3C Recommendation. (See https://www.w3.org/TR/xml11/.)

5.2 Other references

[CEV RDFa]

FHISO (Family History Information Standards Organisation). *Citation Elements: Bindings for RDFa". Exploratory draft of standard. See http://tech.fhiso.org/drafts/rdfa-bindings.

[CEV Vocabulary]

FHISO (Family History Information Standards Organisation). *Citation Elements: Vocabulary". Exploratory draft of standard.

[Chicago]

The Chicago Manual of Style, 16th ed. Chicago: University of Chicago Press, 2010.

[Evidence Explained]

Elizabeth Shown Mills. *Evidence Explained,* 2nd ed. Baltimore: Genealogical Publishing Company, 2009.

[ISO 8601]

ISO (Internation Organization for Standardization). *ISO 8601:2004. Data elements and interchange formats — Information interchange — Representation of dates and times*. 2004.

[JSON-LD]

W3C (World Wide Web Consortium). *JSON-LD 1.0 — A JSON-based Serialization for Linked Data.* Manu Sporny, Gregg Kellogg and Markus Lanthaler, eds., 2014. W3C Recommendation. (See https://www.w3.org/TR/json-ld/.)

[Linked Data]

Heath, Tom and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*, 1st edition. Morgan & Claypool, 2011. (See http://linkeddatabook.com/editions/1.0/.)

[RFC 7159]

IETF (Internet Engineering Task Force). *The JavaScript Object Notation (JSON) Data Interchange Format* Tim Bray, ed., 2014. (See http://tools.ietf.org/html/rfc7159.)

[SWBP XSD DT]

W3C (World Wide Web Consortium). *XML Schema Datatypes in RDF and OWL*. Jeremy J. Carroll and Jeff Z. Pan, 2006. W3C Working Group. See https://www.w3.org/TR/swbp-xsch-datatypes/.

[XML Names]

W3 (World Wide Web Consortium). *Namespaces in XML 1.1*, 2nd edition. Tim Bray, Dave Hollander, Andrew Layman and Richard Tobin, eds., 2006.
W3C Recommendation. See https://www.w3.org/TR/xml-names11/.

[XSD Pt2]

W3 (World Wide Web Consortium). *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. W3C Recommendation. See https://www.w3.org/TR/xmlschema11-2/